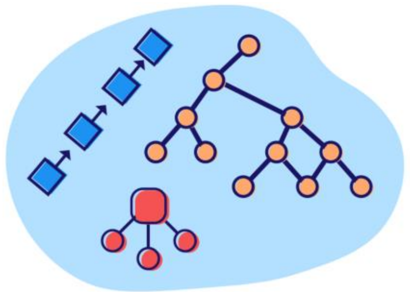
الجامعة العربية الخاصة للعلوم والتكنولوجيا
كلية الهندسة المعلوماتية
قسم تقانة المعلومات

Algorithms & Data Structures (2)

Lecture-6: Binary Trees

(الأشجار الثنائية)



Dr. Asmaa Shaar

فهرس المحاضرة

- الأشجار الثنائية
- أنواع الأشجار الثنائية الخاصة
- تمثيل الأشجار الثنائية
  - باستخدام المصفوفات
  - باستخدام القوائم المترابطة
- أساليب التنقل عبر الأشجار الثنائية

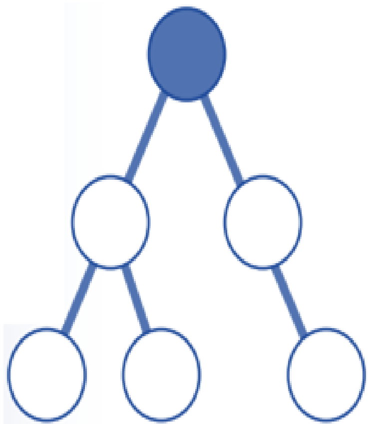
1. أسلوب تنقل VLR ← Pre-order
2. أسلوب تنقل LRV ← post-order
3. أسلوب تنقل LVR ← in-order
4. التنقل بأسلوب level-order

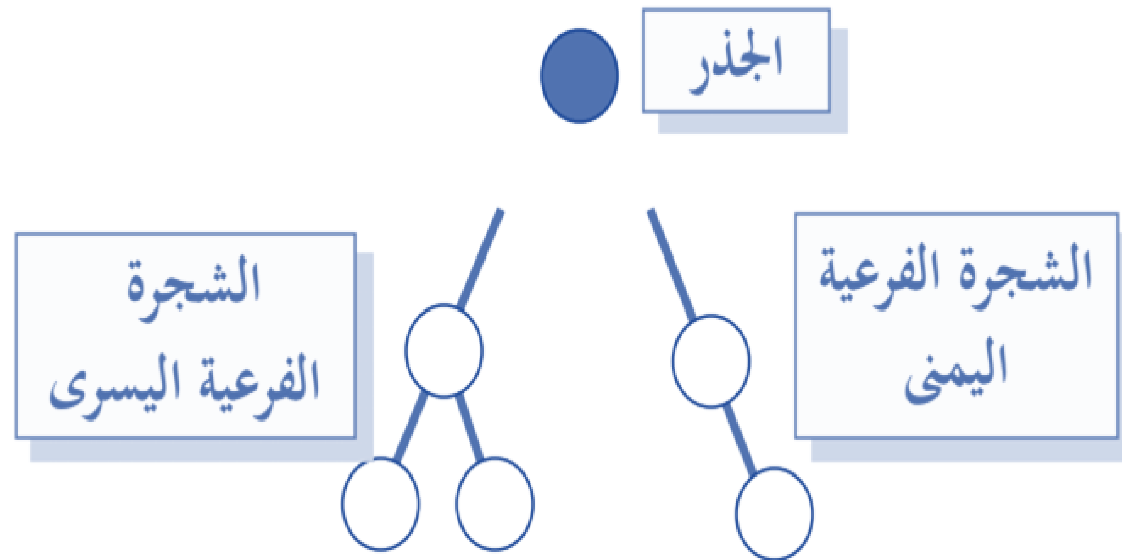
أنواع الأشجار

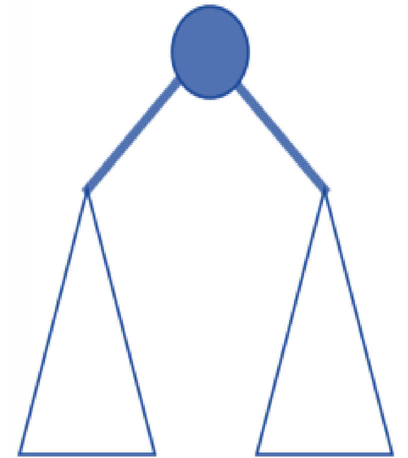
- Binary tree
  - Binary search tree, AVL tree, Red black tree,…
- B tree
- Heaps
- Application Specific tree
- …

الأشجار الثنائية

- نسمي البنية B **شجرة ثنائية** إذا تحقق أحد الشرطين:
  - $B = \emptyset$ الشجرة فارغة.
  - B = <O, B1, B2> حيث Oعقدة الجذر و B1, B2 شجرتان **ثنائيتان** منفصلتان (B1 الشجرة الفرعية اليسارية و B2 الشجرة الفرعية اليمينية).
- بشكل أبسط، هي الأشجار التي يكون فيها لكل عقدة **ابنان على الأكثر.**
  - درجة الشجرة الثنائية=2

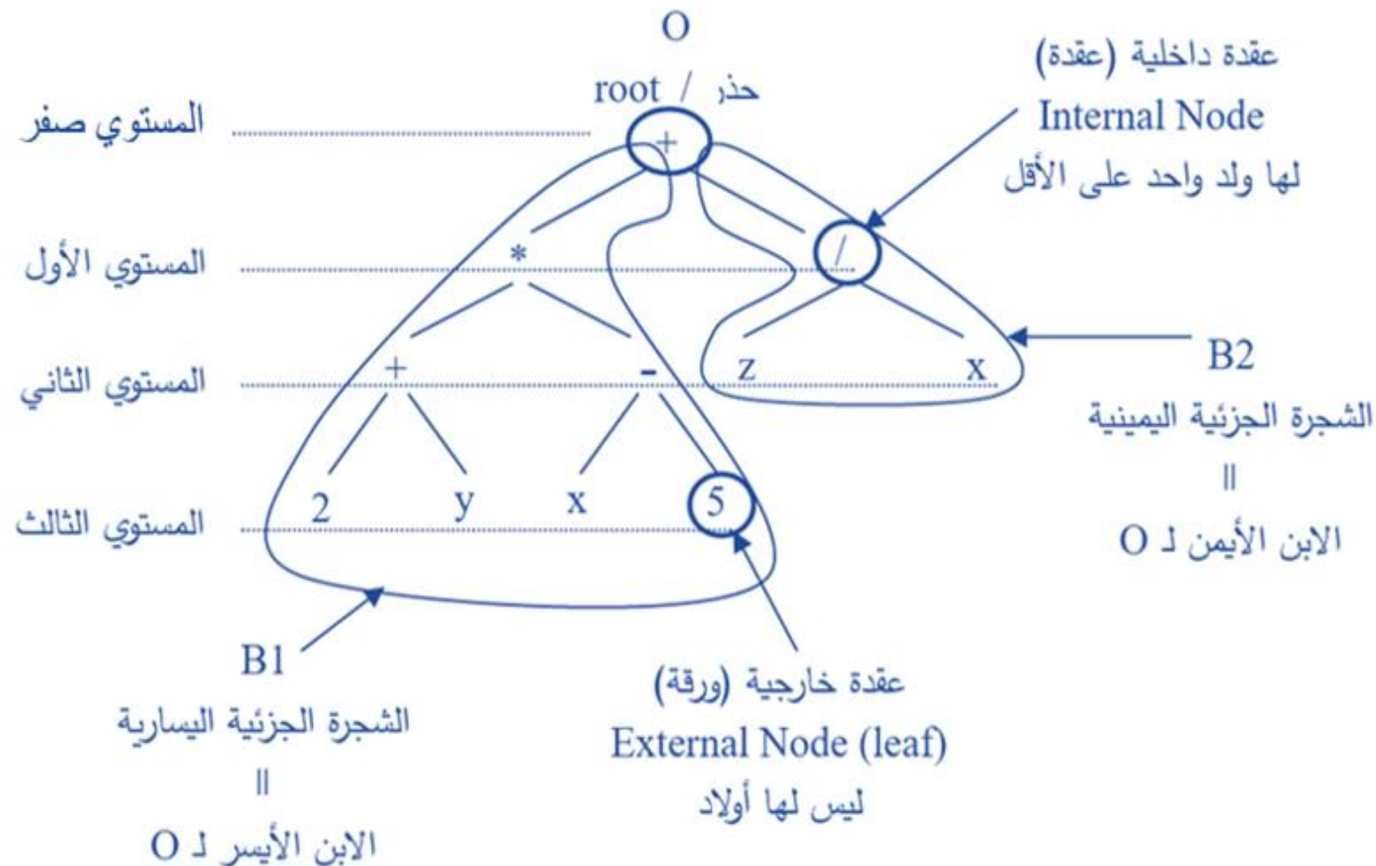






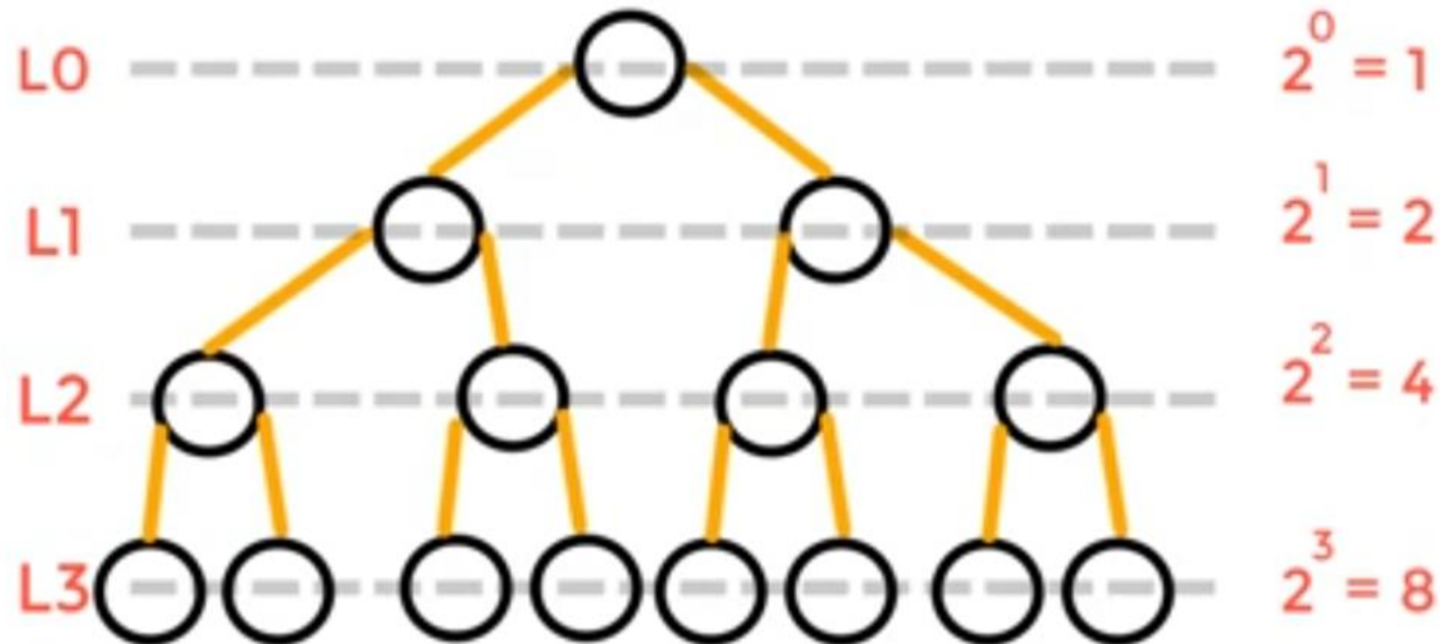
مثال

- مثال: تمثيل تعبير حسابي باستخدام **شجرة ثنائية**: ((2+y)*(x-5))+(z/x)



الأشجار الثنائية

- في الأشجار الثنائية، **العدد الأعظمي** للعقد في المستوي $0 \geq l$ يكون 2^l

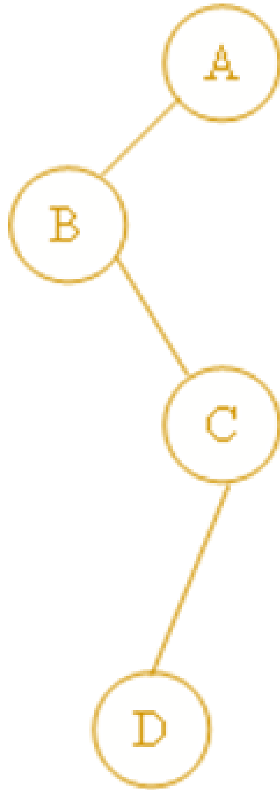


أنواع الأشجار الثنائية الخاصة

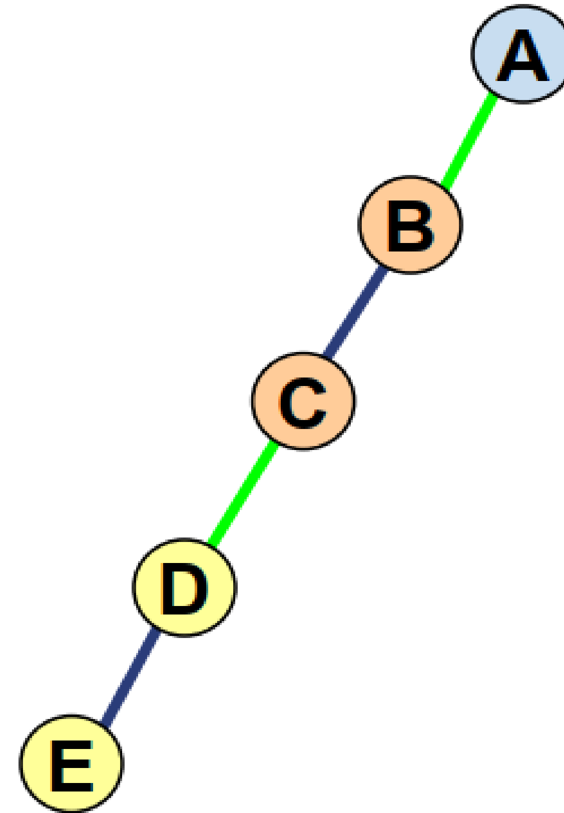
1. الشجرة الثنائية الخطية Linear Binary tree.
2. الشجرة الثنائية الممتلئة Full Binary Tree.
3. الشجرة الثنائية الكاملة Complete Binary Tree .
4. الشجرة الثنائية التامةperfect Binary Tree.

أنواع الأشجار الثنائية الخاصة

- **الشجرة الثنائية الخطية Linear Binary tree:** هي شجرة ثنائية يكون فيها لكل عقدة ولد واحد **على الأكثر**.

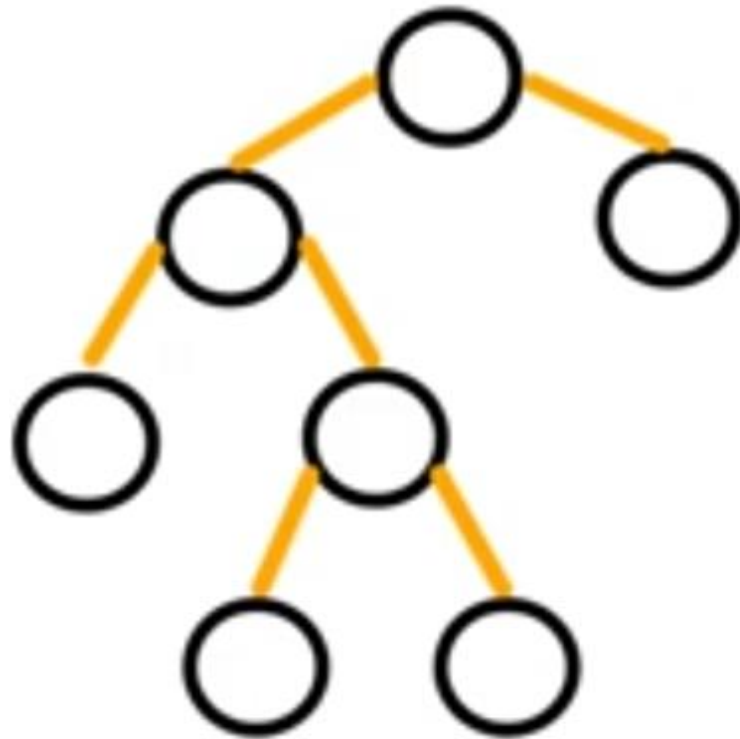


شجرة ثنائية خطية

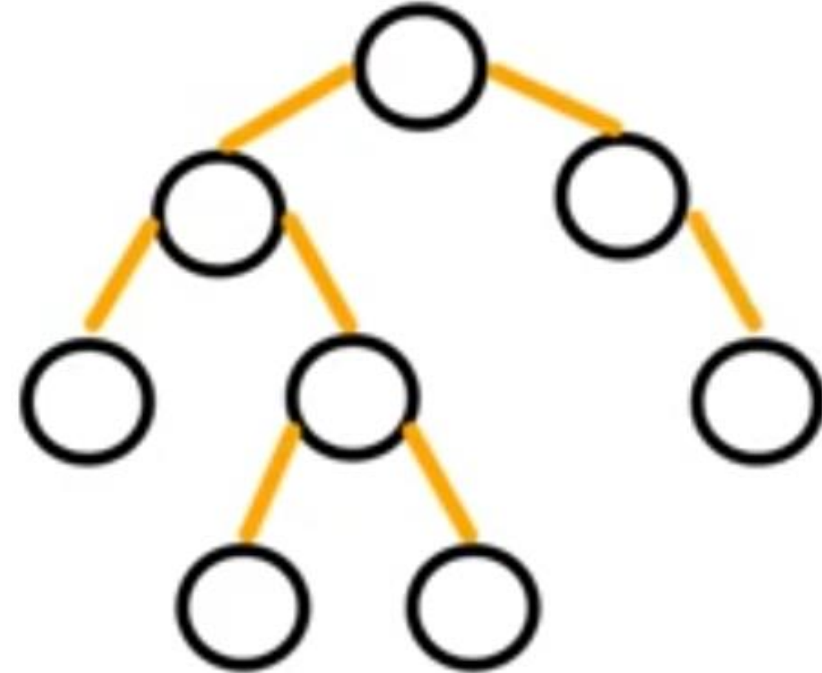


أنواع الأشجار الثنائية الخاصة

- الشجرة الثنائية الممتلئة Full Binary Tree: هي شجرة تمتلك كل عقدة فيها **ولدين بالضبط** أو **صفر ولد** .



Full Binary Tree

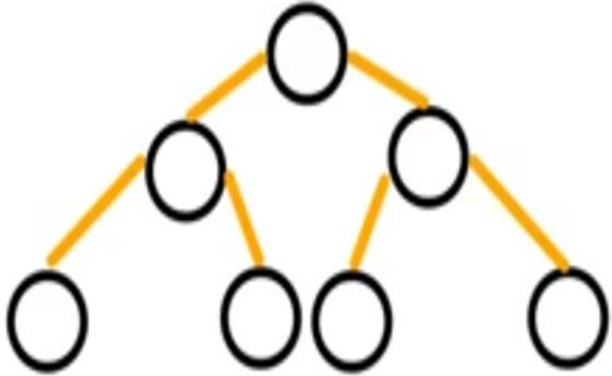


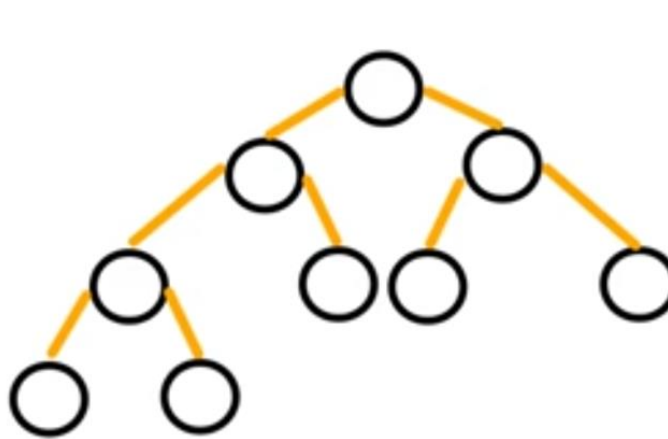
Not Full Binary Tree

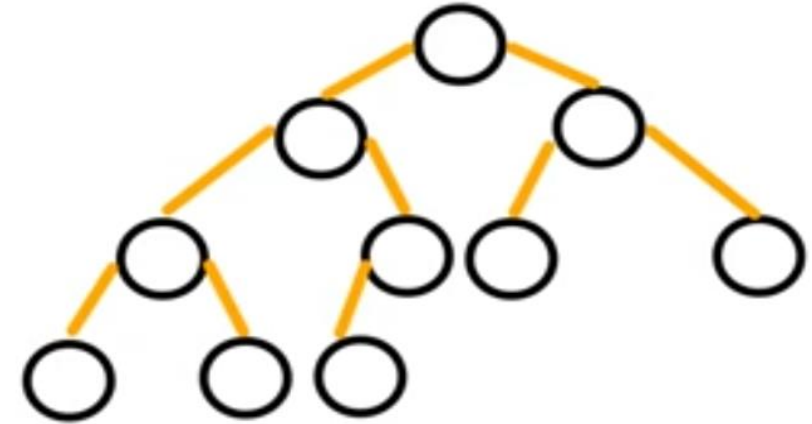
أنواع الأشجار الثنائية الخاصة

- الشجرة الثنائية الكاملة **Complete Binary Tree** : هي شجرة ثنائية جميع مستوياتها ممتلئة بالعقد ماعدا **المستوي الأخير** الذي يمكن أن يحوي فراغات من **جهة اليمين فقط** (بمعنى كل عقد المستوى الأخير تمتد من جهة اليسار لجهة اليمين).

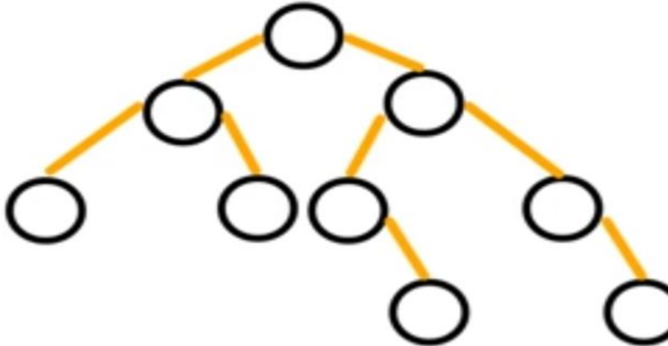
Complete Binary Tree

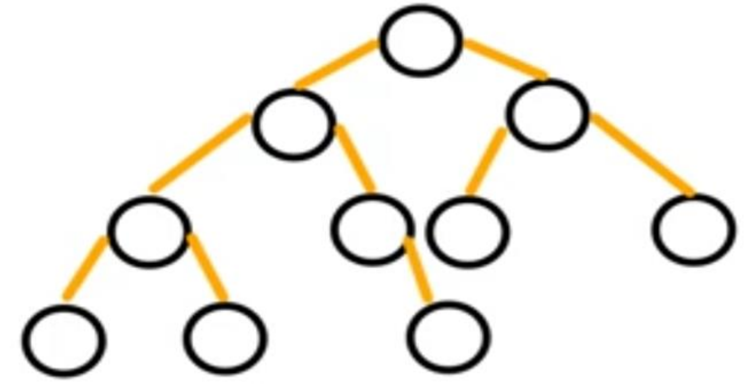






Not Complete Binary Tree





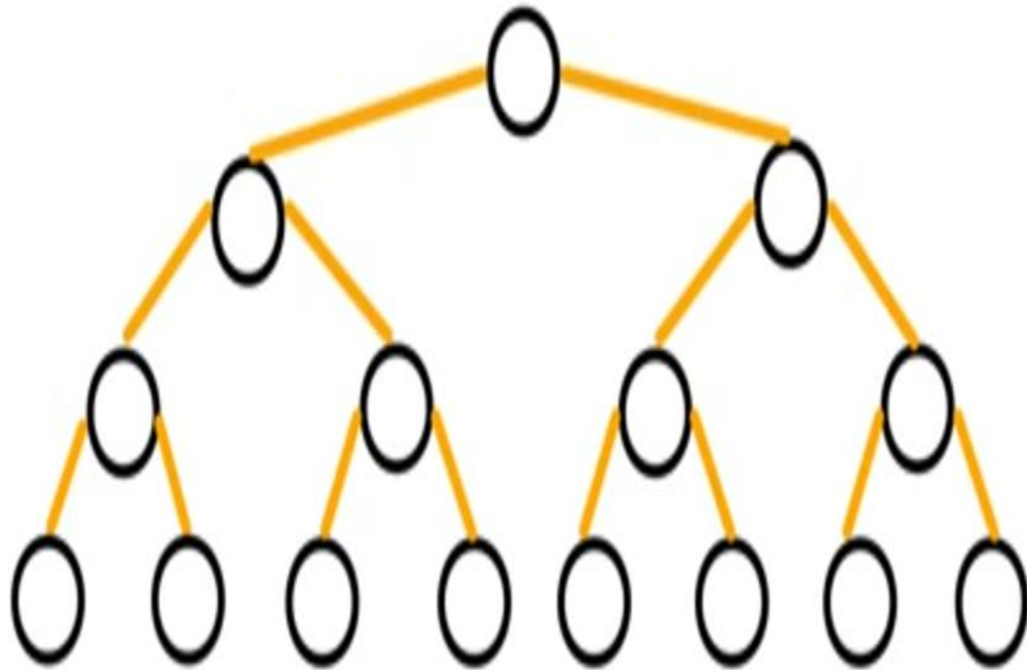
أنواع الأشجار الثنائية الخاصة

- **الشجرة الثنائية التامة perfect Binary Tree:** هي شجرة تكون **جميع** مستوياتها ممتلئة بالعقد ، بمعنى:
  1. كل **عقدة داخلية** تمتلك **بالضبط** ولدين.
  2. كل **العقد الورقية** تحدث في **نفس المستوى**.
- عدد العقد في شجرة تامة ارتفاعها h.

$$1 + 2 + 4 + \ldots\ldots + 2^h = \sum_{k=0}^{k=h} 2^k = 2^{h+1} - 1$$

Find height of tree?!

$$n = 2^{h+1} - 1$$
$$2^{h+1} = (n + 1)$$
$$h = \log_2(n + 1) - 1$$



perfect Binary Tree

تمثيل الأشجار الثنائية

- يمكن تمثيل الأشجار الثنائية بعدة طرق .
  - باستخدام المصفوفات Array Representation
  - باستخدام القوائم المترابطة Linked list Representation

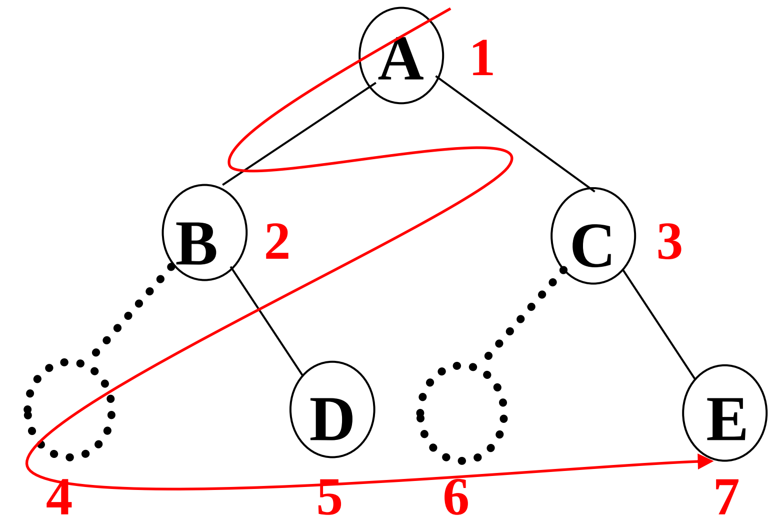
- لكل طريقة **مزاياها** التي تتعلق بالعمليات المطلوب القيام بها على الأشجار .

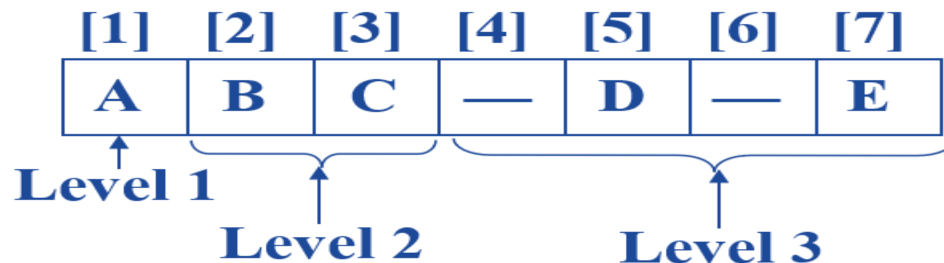
تمثيل الأشجار الثنائية باستخدام المصفوفات

1. باستخدام **مصفوفة أحادية** بسعة (n) مساوية إلى أكبر عدد من العقد في الشجرة الثنائية $(2^{h+1}-1)$.

- من أجل دليل مصفوفة **$1 \le i \le n$** ، فإذا كان i دليل العقدةعندها:

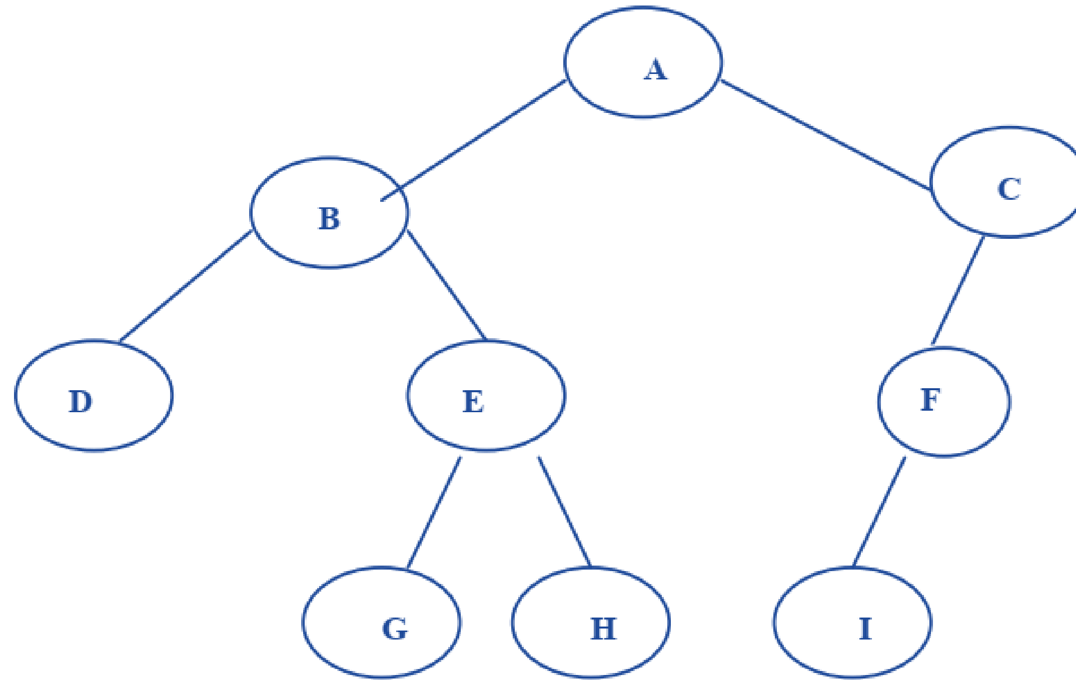
1. **parent(i)** is at $\lfloor i/2 \rfloor$ if **$i \ne 1$**.
   (If i = 1, i is at the root and has no parent).
2. **LeftChild(i)** is at 2i if **$2i \le n$**.
   (If 2i > n, then i has no left child).
3. **RightChild(i)** is at 2i+1 if **$2i+1 \le n$**.
   (If 2i +1 > n, then i has no right child).





تمثيل الأشجار الثنائية باستخدام المصفوفات

1. باستخدام **مصفوفة أحادية**.
   - مثال1:



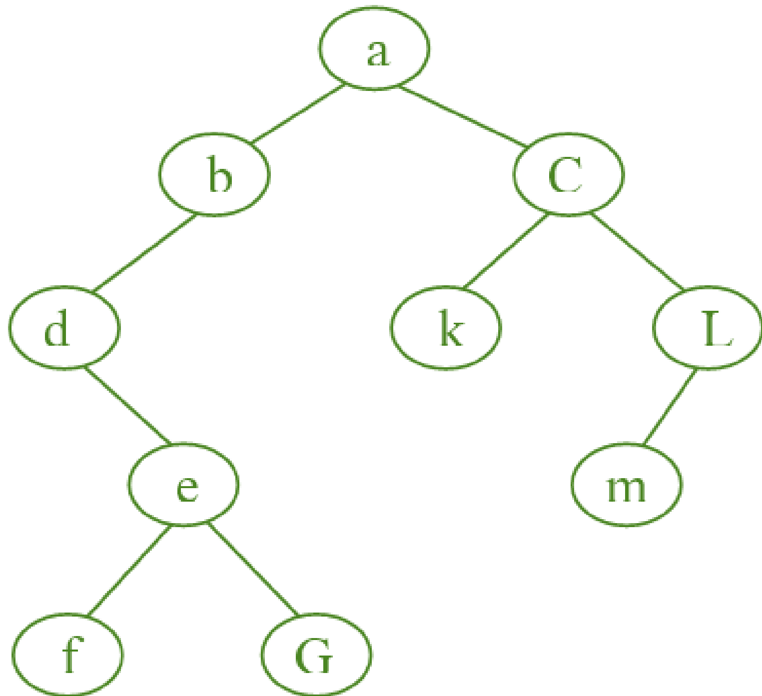
اولاً:- نجد ارتفاع الشجرة (h=3)

ثانياً :- نجد اكبر عدد من العقد ممكن ان يخزن في تلك الشجرة $2^{h+1}-1=2^{3+1}-1=16-1=15$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D	E	F	-	-	-	G	H	I	-	-	-

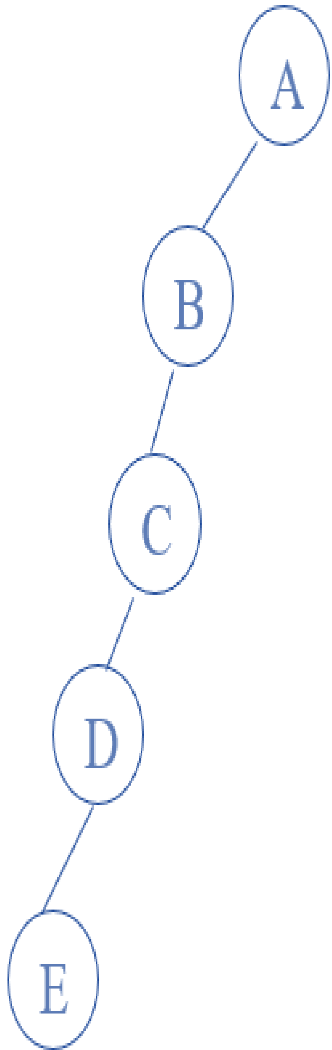
تمثيل الأشجار الثنائية باستخدام المصفوفات

2. باستخدام **مصفوفة ثنائية**: نعرف مصفوفة فيها ثلاثة أعمدة بحيث يحوي كل سطر فيها معلومات العقدة (العمود الاول)، دليل الابن اليساري (العمود الثاني) و دليل الابن اليميني (العمود الثالث).



1	a	2	3
2	b	4	-
3	C	6	7
4	d	-	9
5	-	-	-
6	k	-	-
7	L	14	-
8	-	-	-
9	e	18	19
-	-	-	-
14	m	-	-
-	-	-	-
18	f	-	-
19	G	-	-

تمثيل الأشجار الثنائية باستخدام المصفوفات



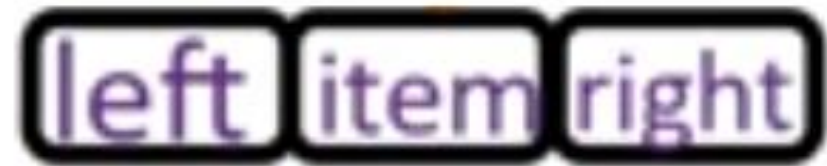
[1]	A
[2]	B
[3]	--
[4]	C
[5]	--
[6]	--
[7]	--
[8]	D
[9]	--
.	.
[16]	E
.	.
[31]	

- التمثيل باستخدام المصفوفات يكون مقبولا **للأشجار الثنائية <u>الممتلئة بالعقد</u>**، ولكنه قد يؤدي إلى **هدر لجزء من المساحة التخزينية (الذاكرة)** من أجل تمثيل الأنواع الأخرى من الأشجار الثنائية.
  - في أسوأ حالة، تحتاج شجرة خطية بارتفاع h إلى مصفوفة بسعة $2^{h+1} - 1$ بحيث يتم استخدام $(h+1)$ خانة فقط.
- إضافة إلى ذلك، إن حذف أو إضافة عقد من/إلى العقد الداخلية في الشجرة يتطلب **تحريك عدة عقد** مما يؤدي إلى **تغيير ادلة بعض العقد** المخزنة في المصفوفة .

تمثيل الأشجار الثنائية باستخدام القوائم المترابطة

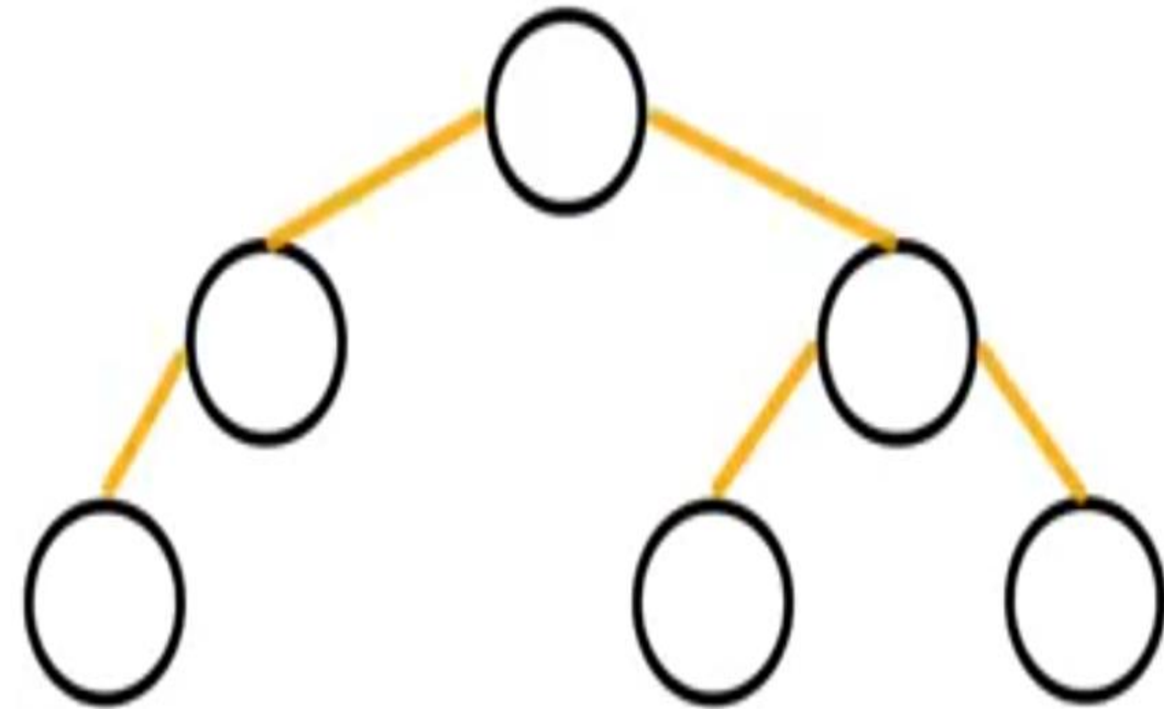
- يتم تمثيل كل عقدة بسجل يمتلك ثلاثة حقول :
  - item: لتخزين المحتوى الفعلي للعقدة.
  - left: مؤشر إلى الشجرة الجزئية اليسارية
  - right:مؤشر إلى الشجرة الجزئية اليمينية
- و تكون الشجرة عبارة عن مؤشر إلى العقدة الأولى (أي الجذر) .

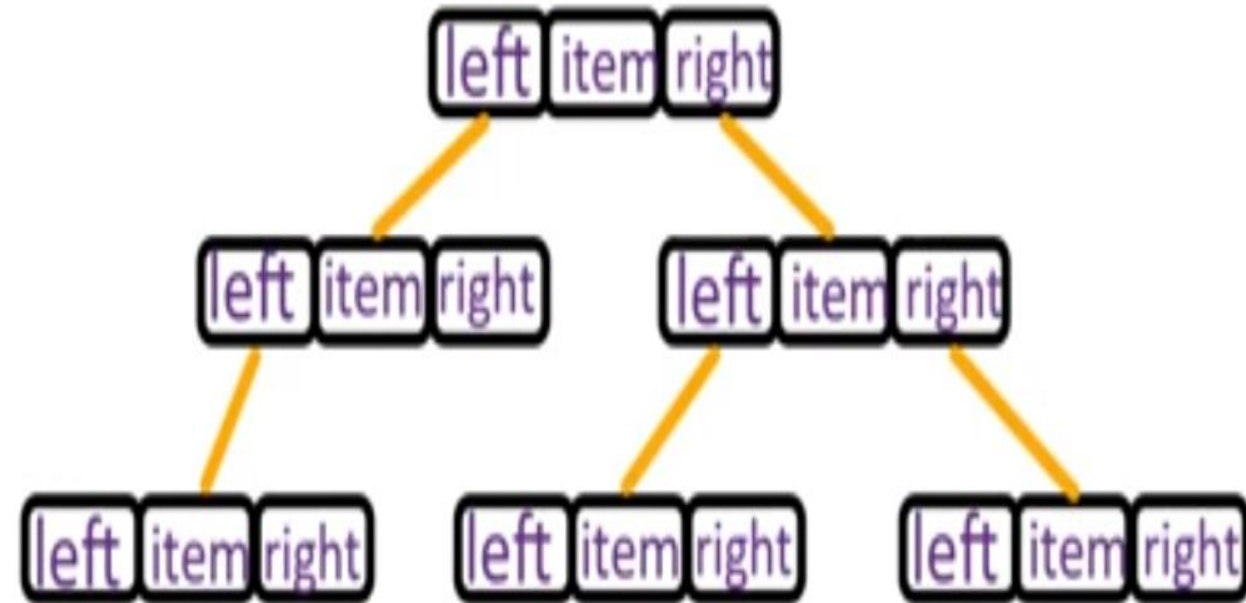
```
struct node {
      char item; //int,char,…
      node *left, *right;
                     };
node *root;
```



تمثيل الأشجار الثنائية باستخدام القوائم المترابطة

- مثال:





أساليب التنقل عبر الأشجار الثنائية

Binary Trees Traversing

أساليب التنقل عبر الأشجار الثنائية

- يقصد بعملية التنقل، المرور أو زيارة كل عقدة من عقد الشجرة حسب **ترتيب معين** بهدف **إجراء عملية معالجةٍ ما** عليها (على سبيل المثال، طباعة، بحث، حشر، حذف،....).
- نرمز للتحرك يساراً بـ **L** وللتحرك يميناً بـ **R** وللمرور بالعقدة (زيارة العقدة) بـ **V** ← وبالتالي يوجد ستة تراكيب هي:

RVL, RLV, VRL , LVR, LRV, VLR

- بفرض أنه يجب التحرك يساراً قبل التحرك يميناً، عندئذ يبقى فقط ثلاثة تراكيب :

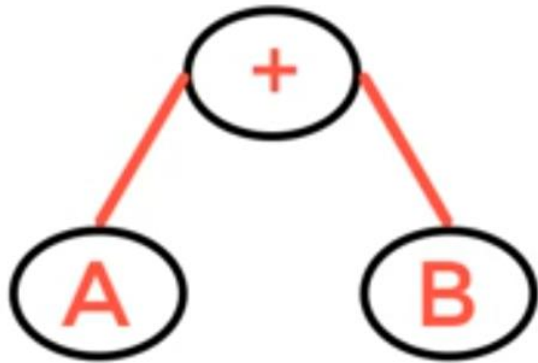
LVR, LRV, VLR

- واعتماداً على موقع V بالنسبة لـ L و R ،تدعى الأساليب السابقة بـ :

1. أسلوب تنقل Pre–order ← VLR
2. أسلوب تنقل post–order ← LRV
3. أسلوب تنقل in–order ← LVR

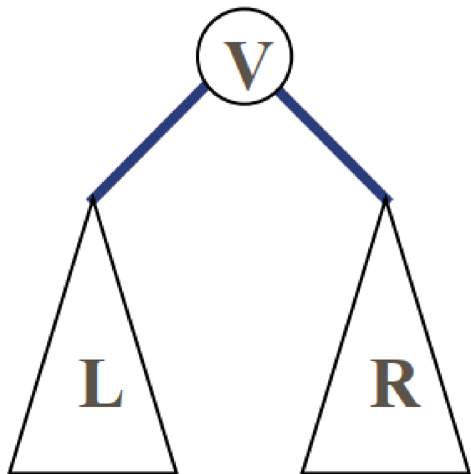
التنقل بأسلوب In–order

- يبدأ التنقل بزيادة الابن الايسر (L) ثم العقدة (V) ثم الابن الأيمن (R).
- نرمز له بـ (LVR) أو (left root right).
- مثال: قم بطباعة عقد الشجرة التالية لدى زيارتها بأسلوب in–order



In-order : left root right

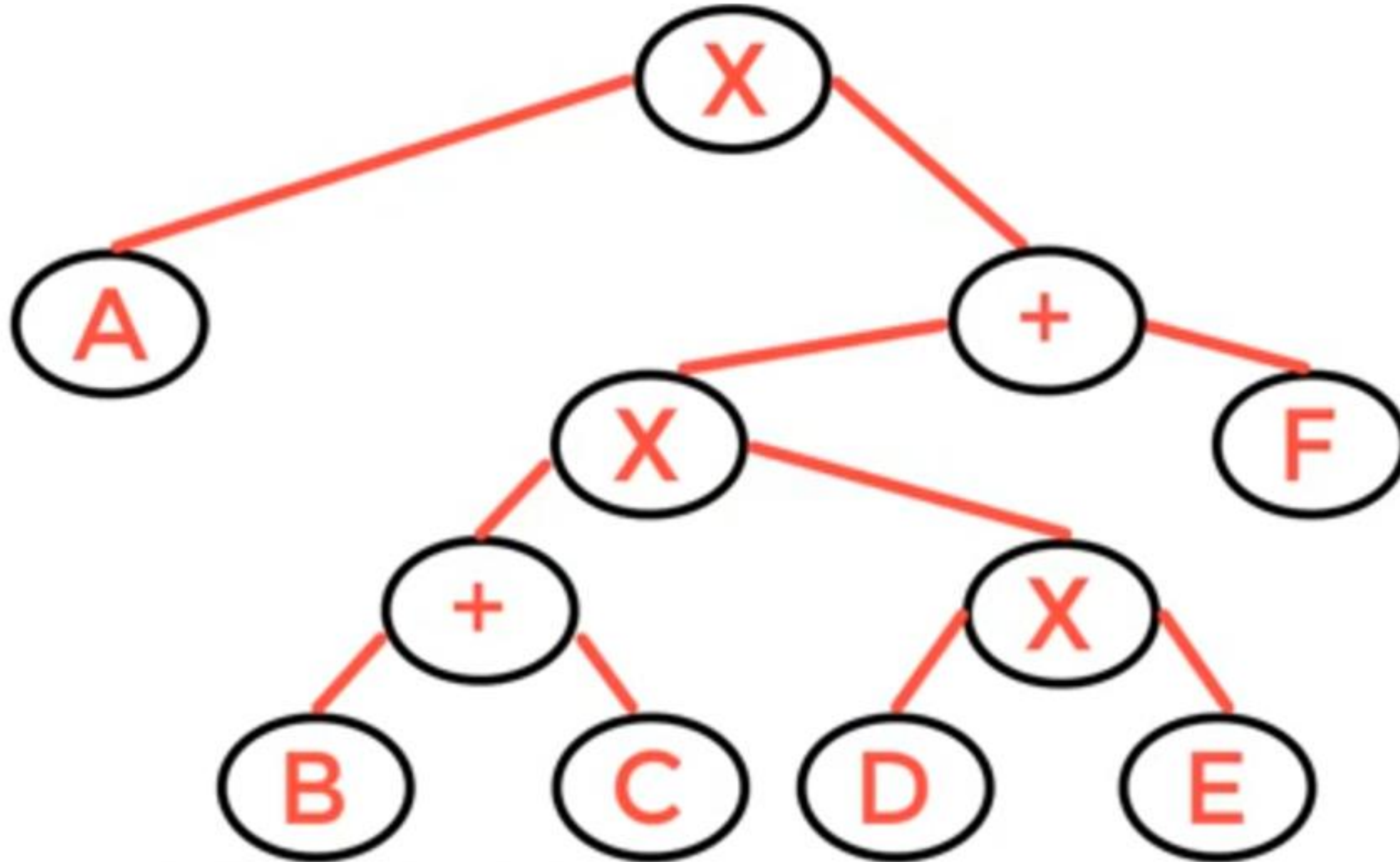
A + B



- عملياً يتم زيارة جميع العقد في الشجرة الفرعية اليسارية (Left sub–tree) بأسلوب in–order ثم العقدة الاب (V) ثم جميع العقد في الشجرة الفرعية اليمينية (Right sub–tree) بأسلوب in–order.

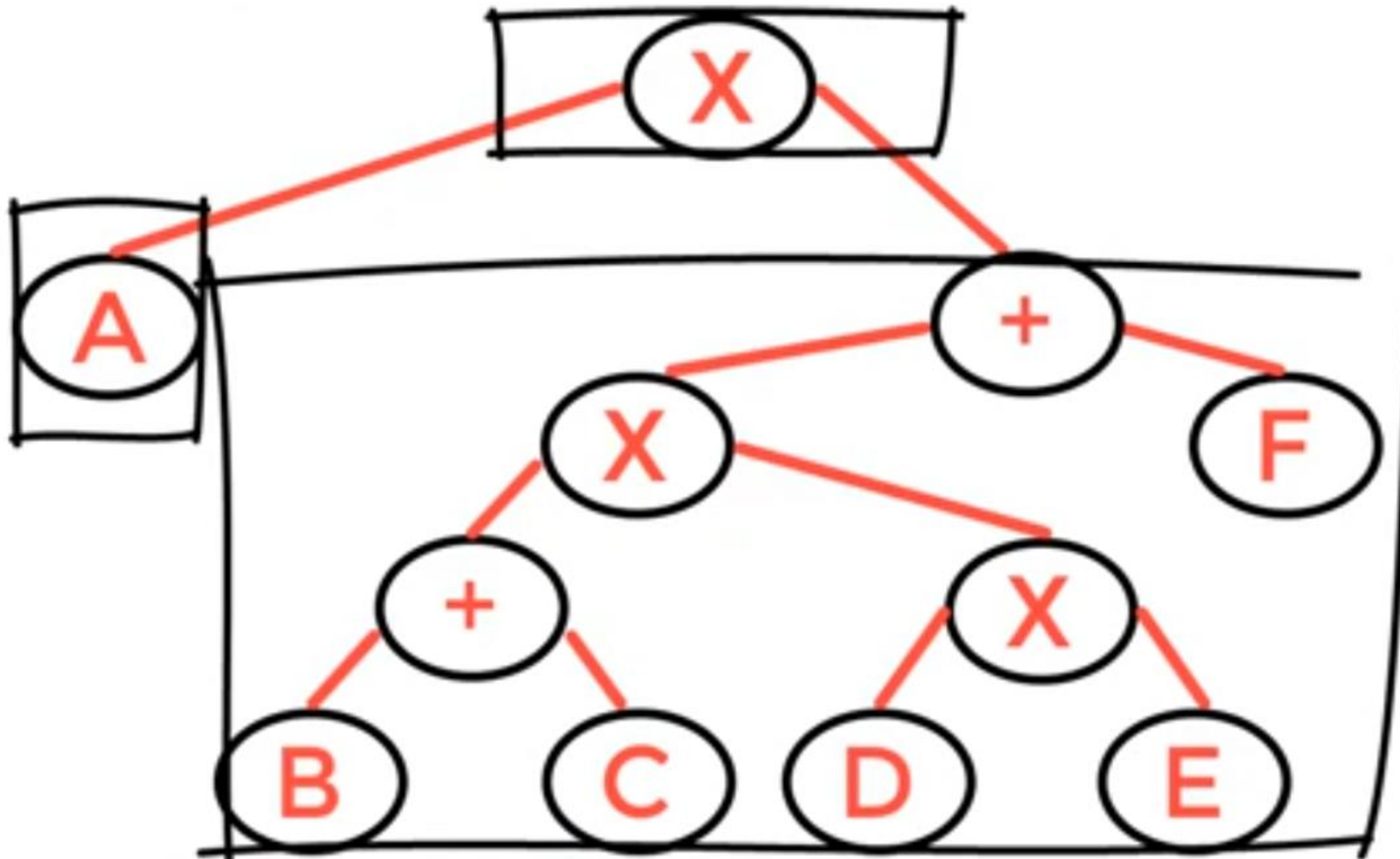
التنقل بأسلوب In-order

- مثال:



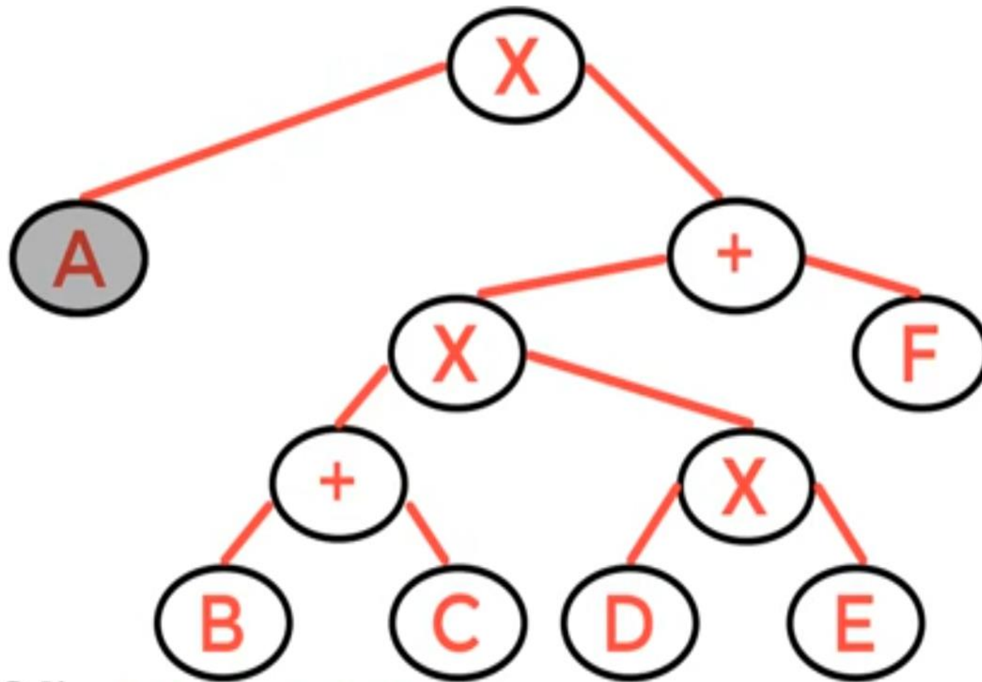
التنقل بأسلوب In-order

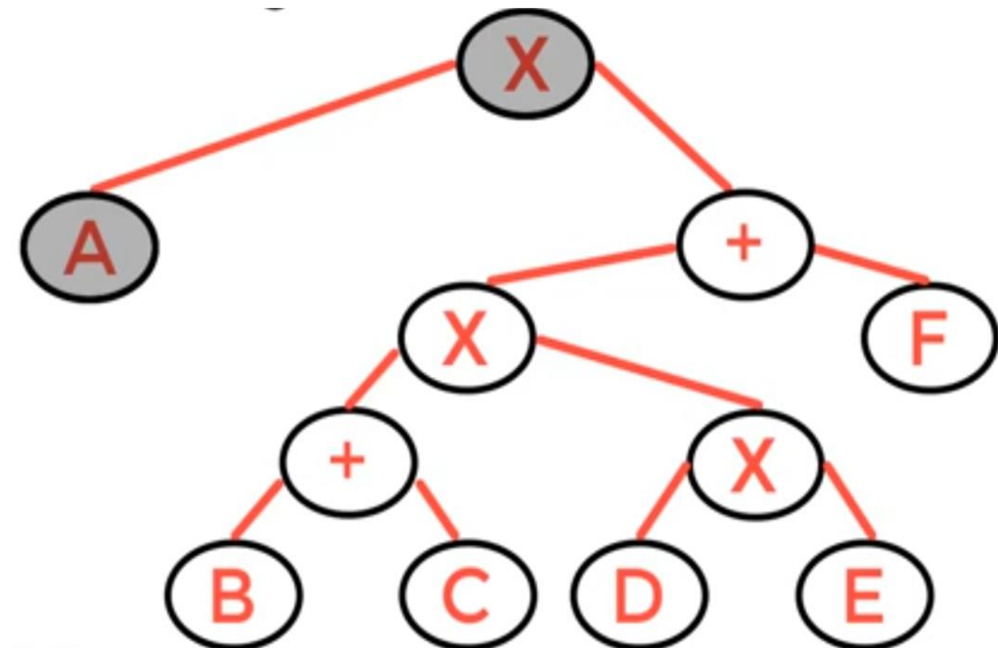
- مثال:



التنقل بأسلوب In-order

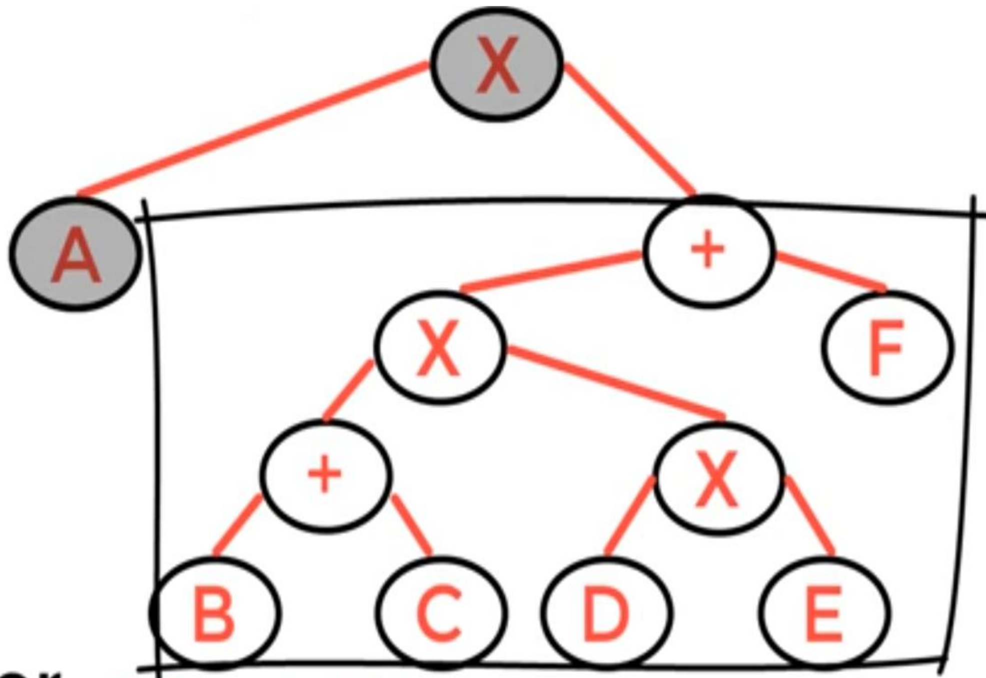
- مثال:

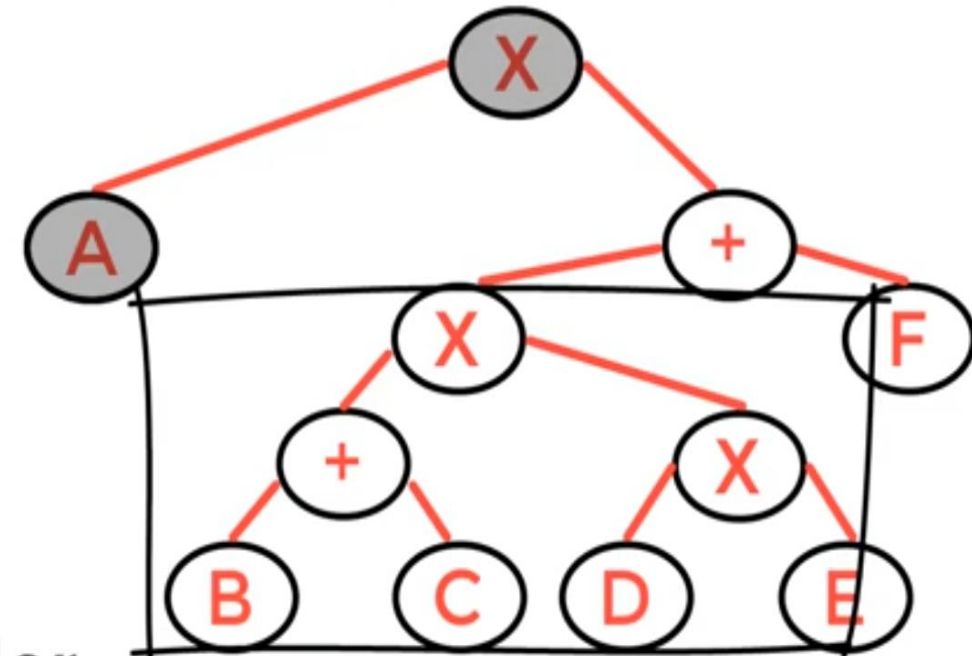




التنقل بأسلوب In–order

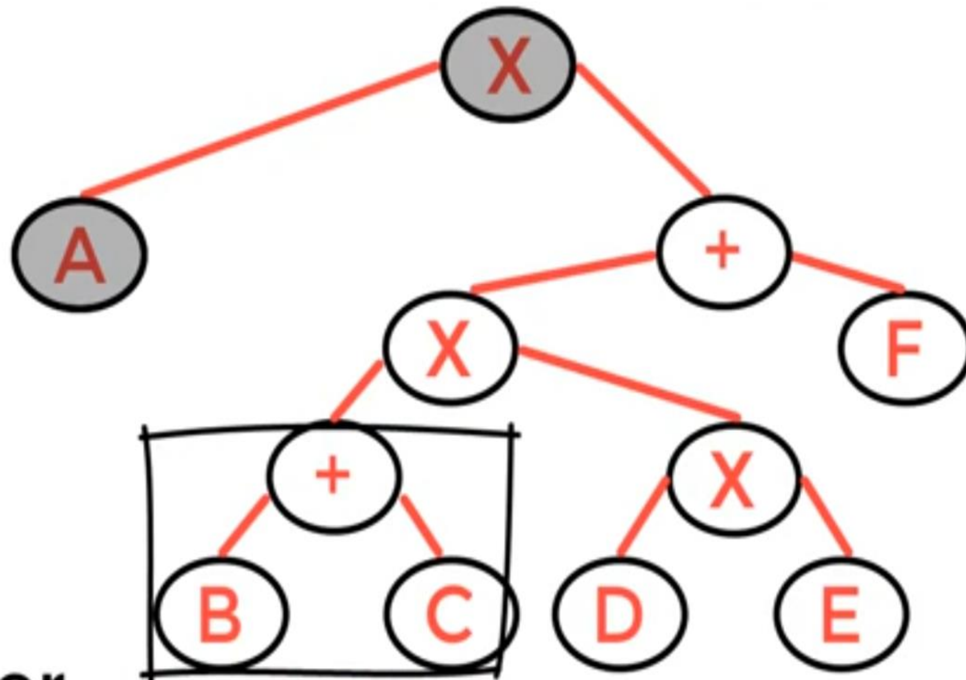
- مثال:

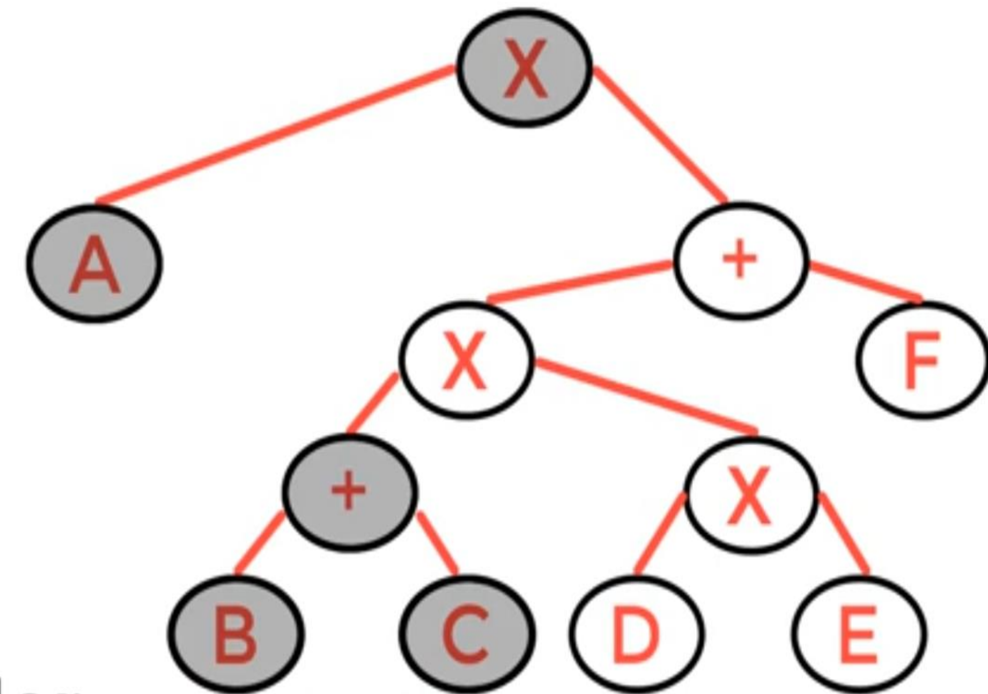




التنقل بأسلوب In–order

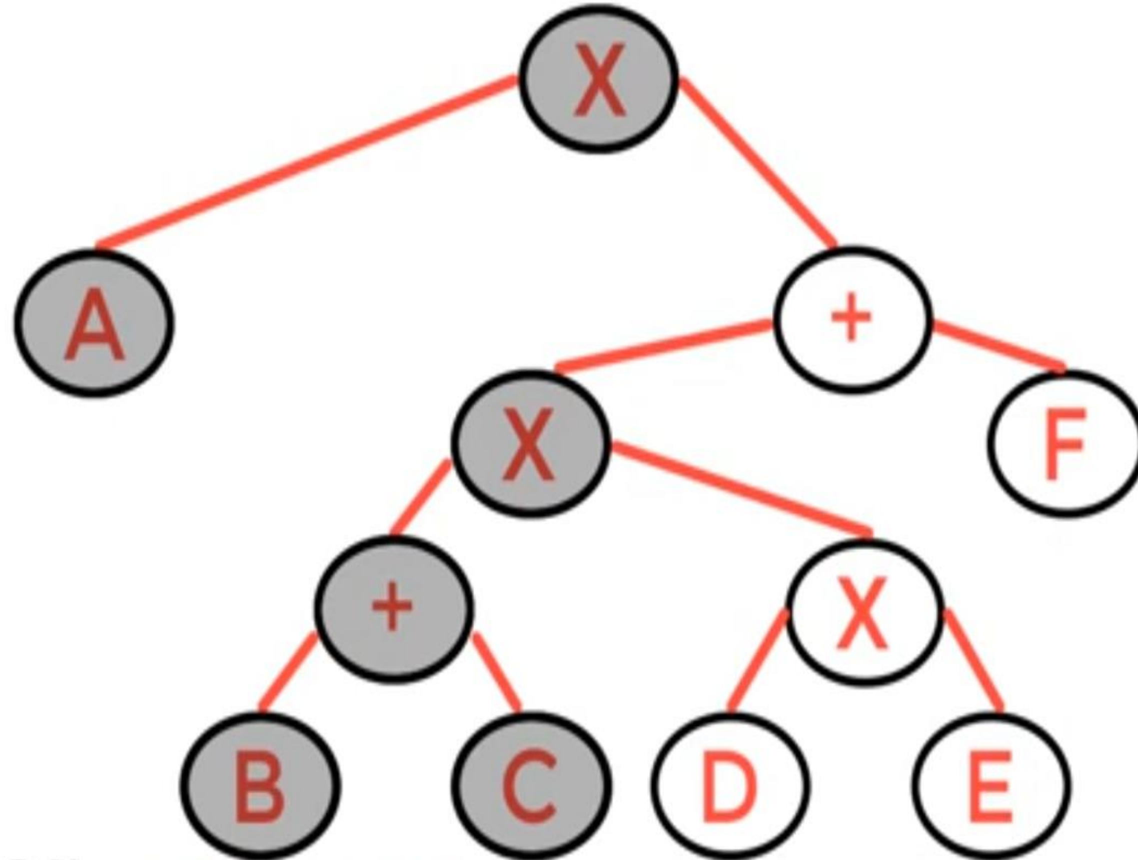
- مثال:





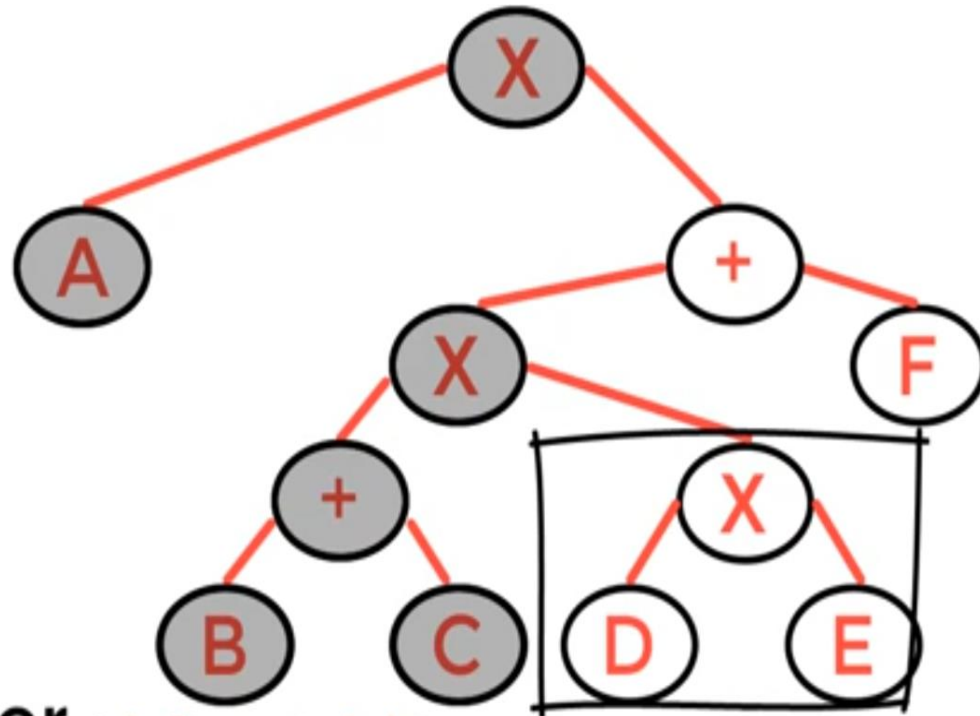
التنقل بأسلوب In-order

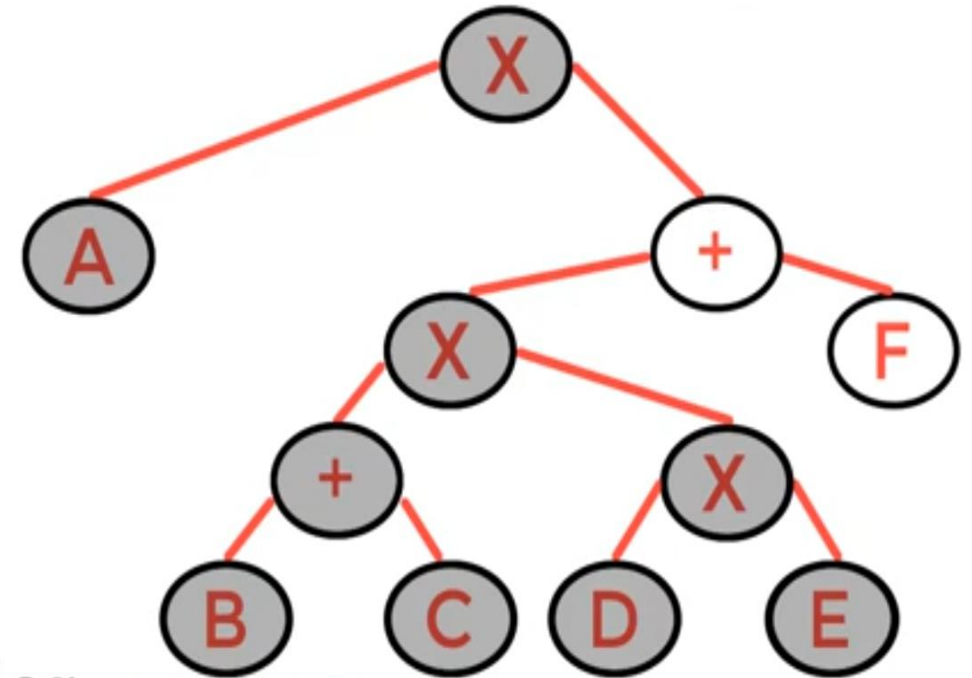
- مثال:



التنقل بأسلوب In-order

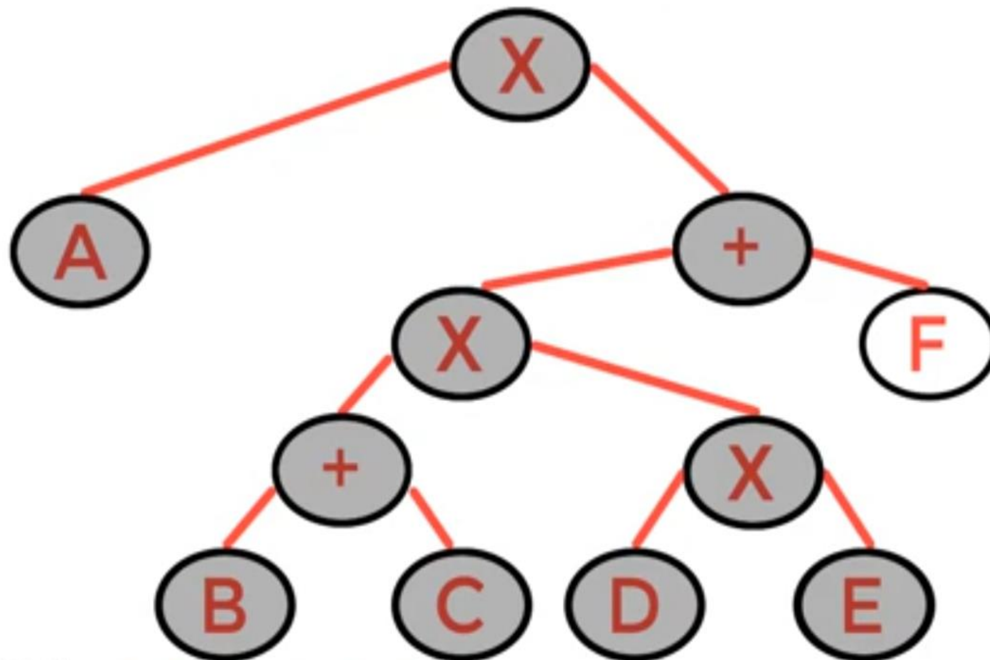
- مثال:

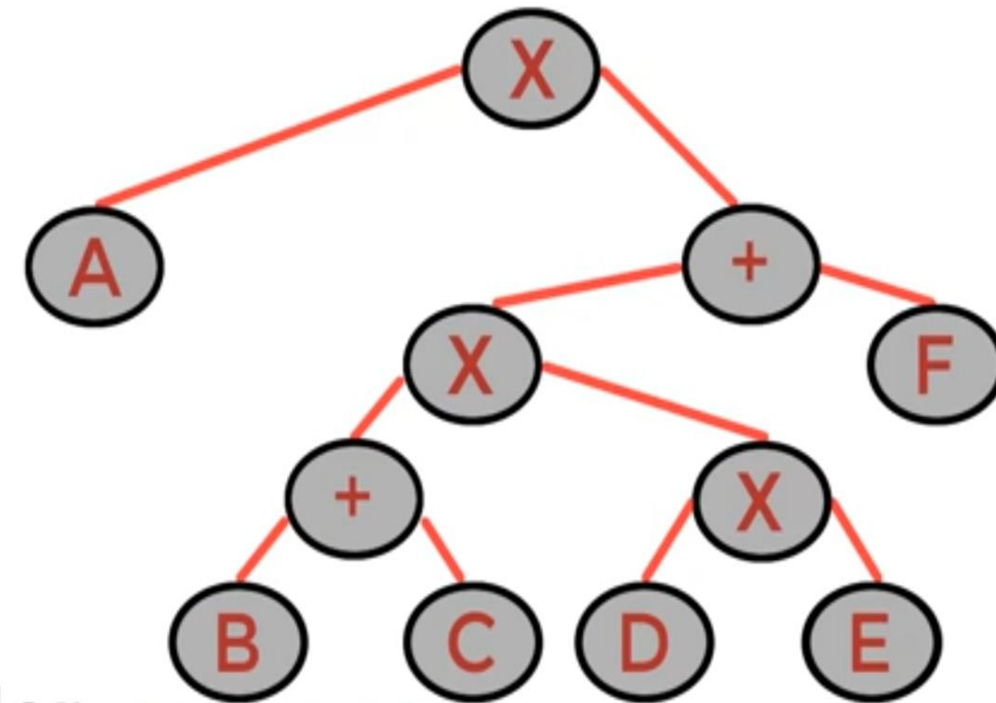




التنقل بأسلوب In–order

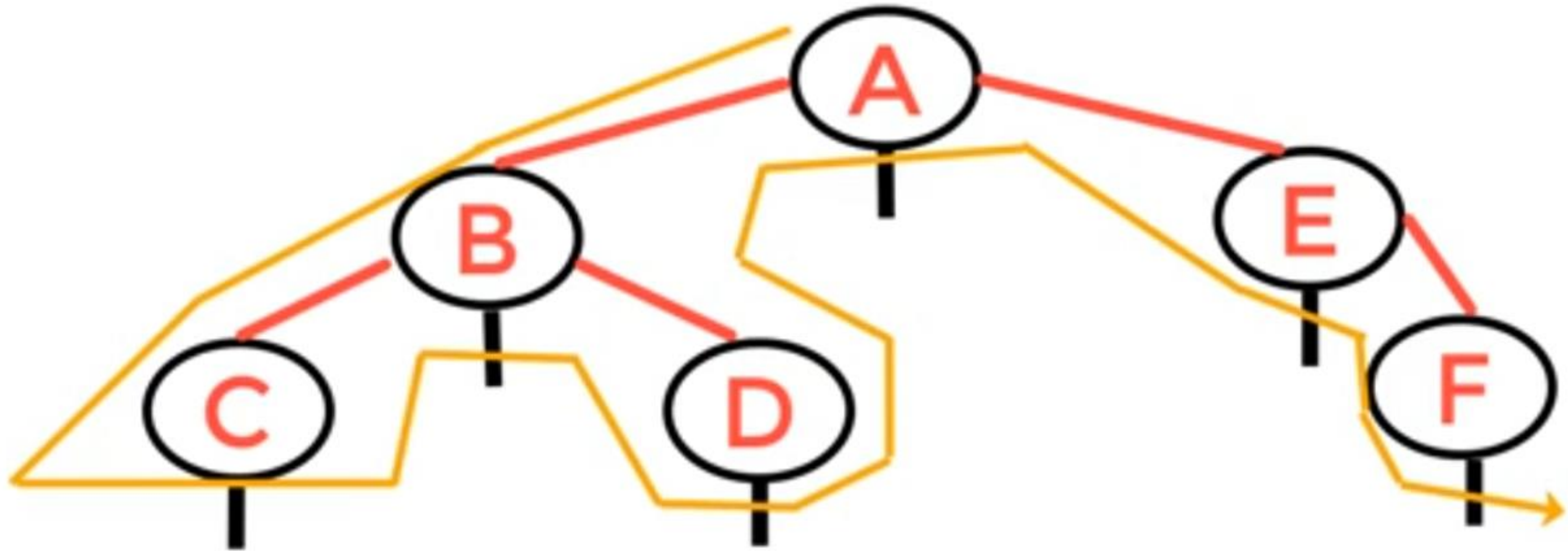
- مثال:





التنقل بأسلوب In–order

- مثال:

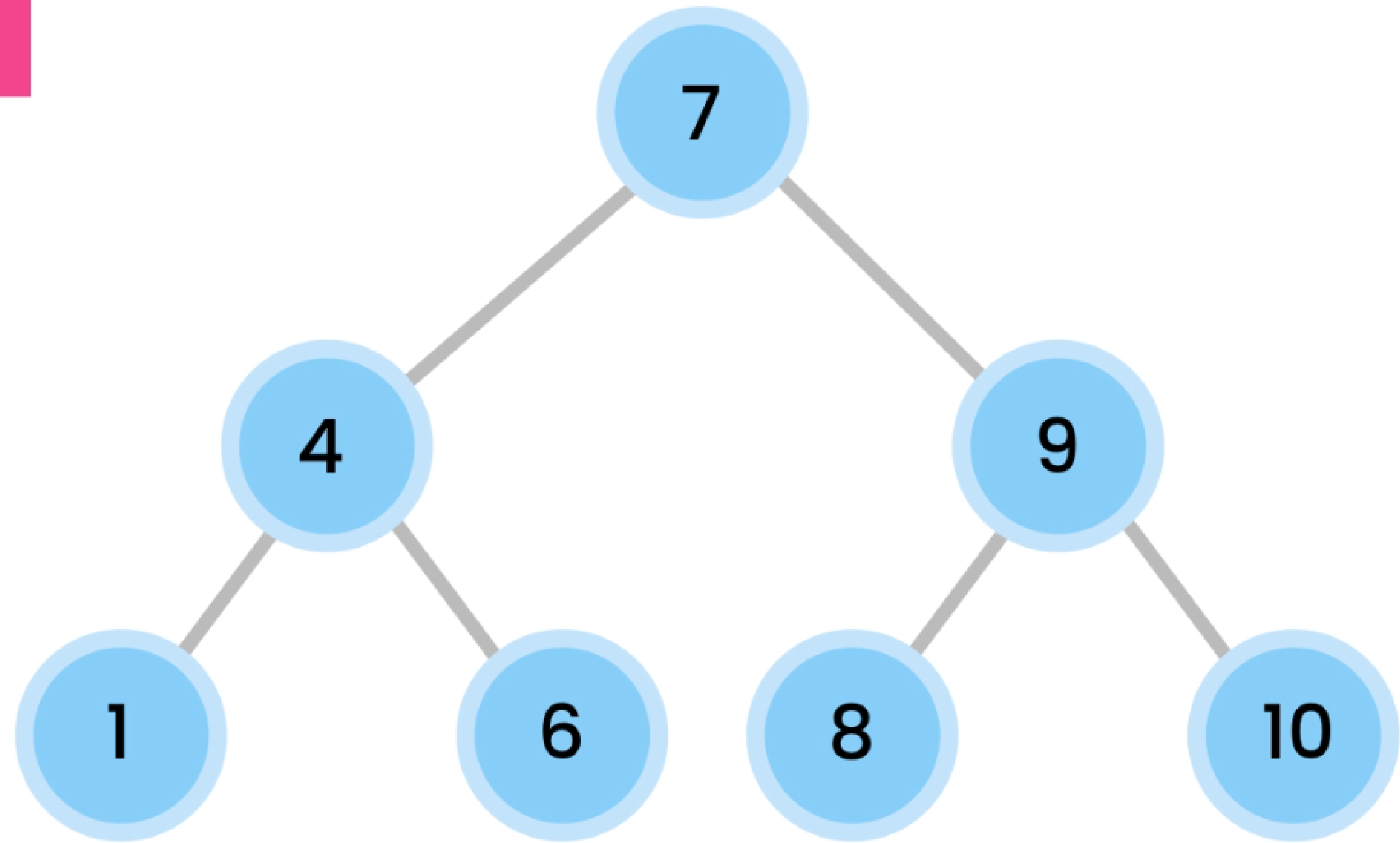


التنقل بأسلوب In-order

- مثال:

IN-ORDER

Left, Root, Right



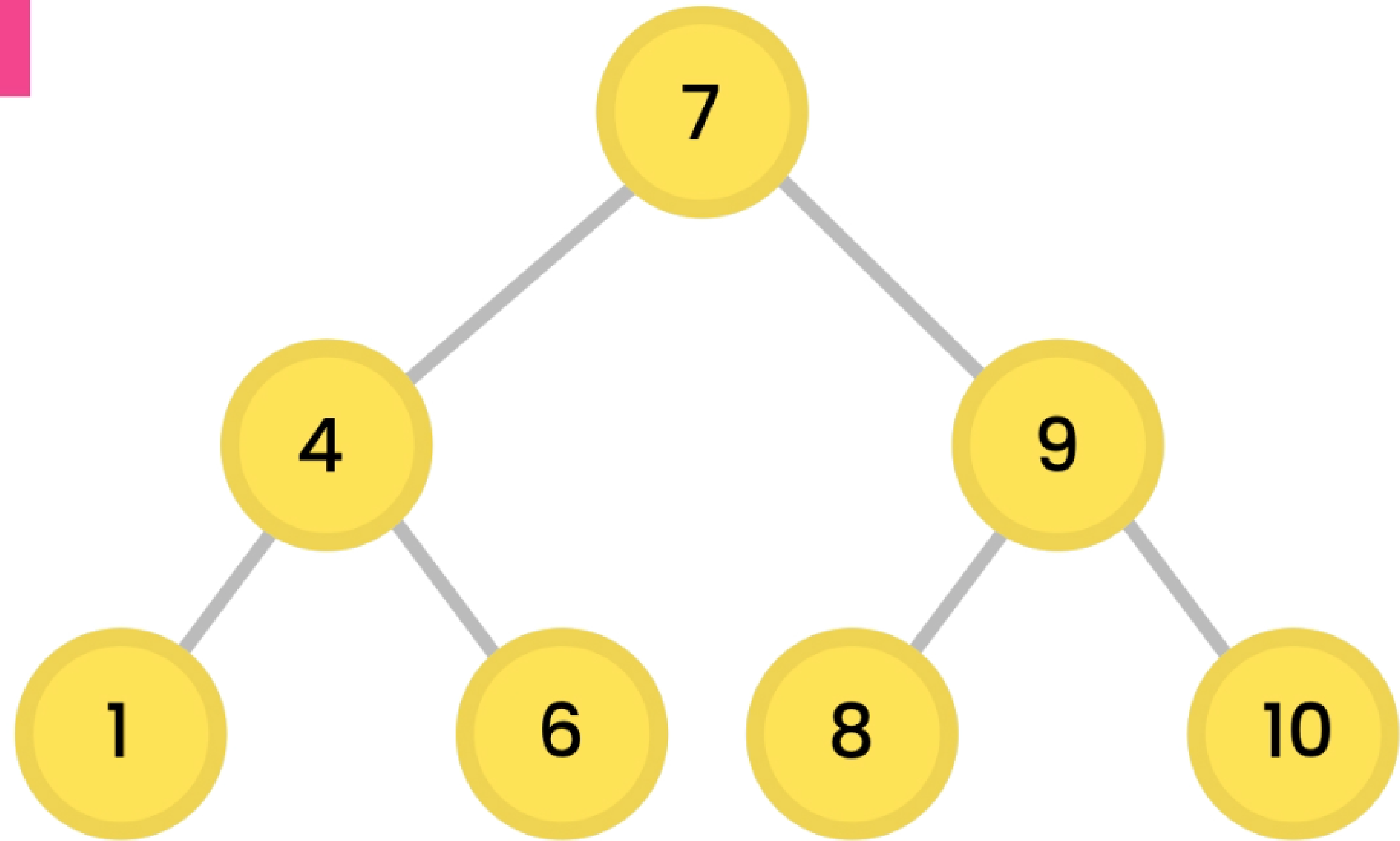
التنقل بأسلوب In–order

- مثال:

IN-ORDER

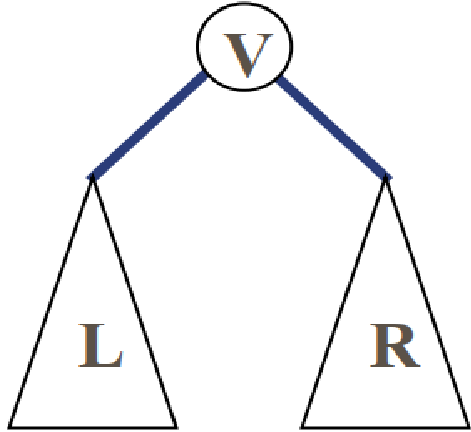
Left, Root, Right

1, 4, 6, 7, 8, 9, 10



التنقل بأسلوب In–order

- يمكننا استخدام مفهوم العودية في كتابة الدالة الإجرائية التي تنفذ التنقل بأسلوب In–order، وستؤدي هذه الدالة ثلاث مهام:
  1. تستدعي نفسها عودياً للتنقل عبر الشجرة الفرعية اليسارية للعقدة **V**
  2. تقوم بزيارة العقدة **V**
  3. تستدعي نفسها عودياً للتنقل عبر الشجرة الفرعية اليمينية للعقدة **V**

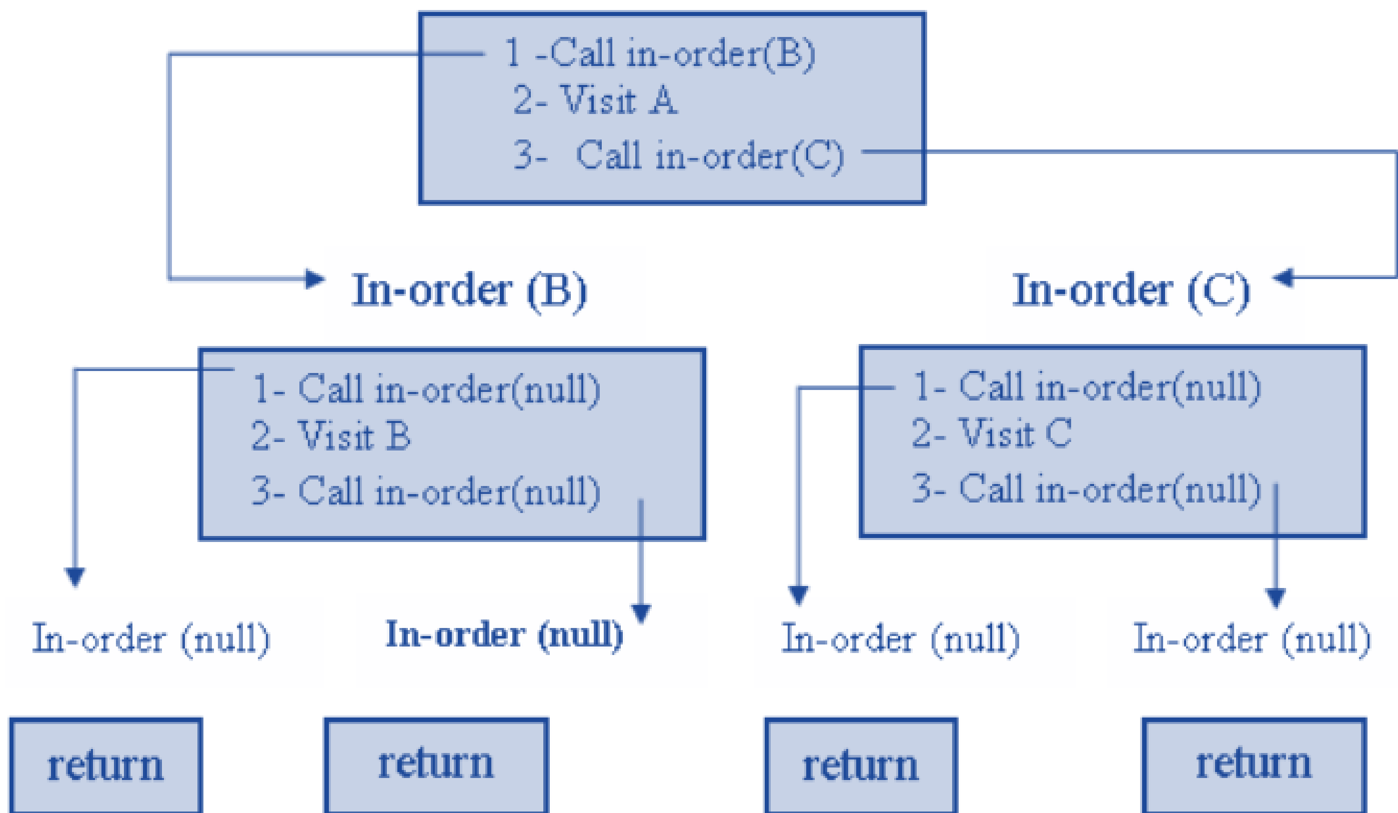


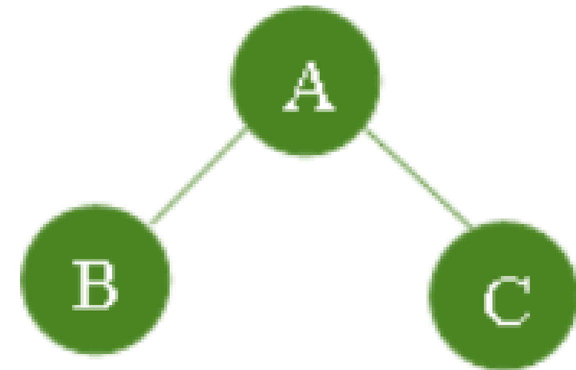

```
void inOrder(node *p)
        {
         if (p!=Null){
            inOrder(p->left);
            cout<< p->item<<" ";
            inOrder(p->right);
               }
            }
```

left	item	right

التنقل بأسلوب In-order

- مثال يوضح آلية العمل للدالة العودية inOrder() باستخدام شجرة مكونة من ثلاث عقد.





التنقل بأسلوب In-order

- باستخدام مفهوم المكدس ، يمكننا كتابة الدالة in-order **بالصيغة التكرارية** حسب الخطوات التالية:

1) Create an empty stack S.
2) Initialize current=root
3) Push the current node to S and set current = current->left
   until current is NULL
4) If current is NULL and stack is not empty then
   a) Pop the top item from stack.
   b) Print the popped item
   c) set current = popped_item->right and Go to step 3.
5) If current is NULL and stack is empty then we are done.

التنقل بأسلوب In-order

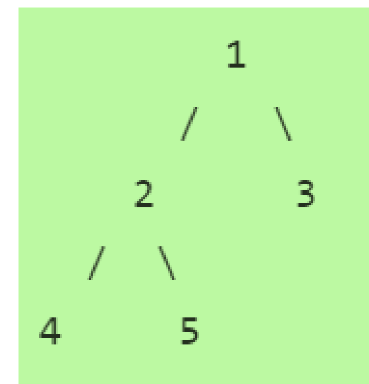
- مثال:

Step 1 Creates an empty stack: S = NULL

Step 2 sets current as address of root: current -> 1

Step 3 Pushes the current node and set current = current->left until current is NULL

push 1: Stack S -> 1
current -> 2
push 2: Stack S -> 2, 1
current -> 4
push 4: Stack S -> 4, 2, 1
current = NULL



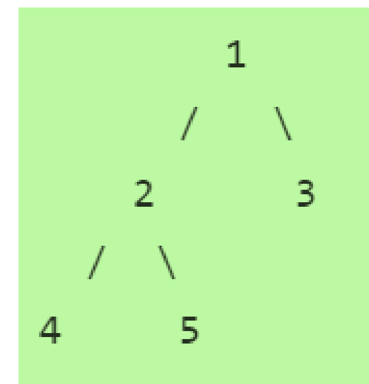
التنقل بأسلوب In-order

- مثال:

```
Step 4 pops from S
     a) Pop 4: Stack S -> 2, 1
     b) print "4"
     c) current = NULL /*right of 4 */ and go to step 3
Since current is NULL step 3 doesn't do anything.

Step 4 pops again.
     a) Pop 2: Stack S -> 1
     b) print "2"
     c) current -> 5/*right of 2 */ and go to step 3

Step 3 pushes 5 to stack and makes current NULL
     Stack S -> 5, 1
     current = NULL
```



التنقل بأسلوب In-order

- مثال:

Step 4 pops from S

a) Pop 5: Stack S -> 1

b) print "5"

c) current = NULL /*right of 5 */ and go to step 3

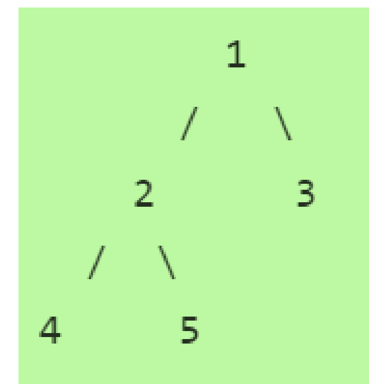
Since current is NULL step 3 doesn't do anything

Step 4 pops again.

a) Pop 1: Stack S -> NULL

b) print "1"

c) current -> 3 /*right of 1 */



التنقل بأسلوب In-order

- مثال:

Step 3 pushes 3 to stack and makes current NULL

Stack S -> 3

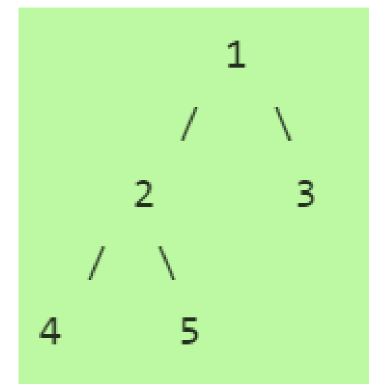
current = NULL

Step 4 pops from S

a) Pop 3: Stack S -> NULL

b) print "3"

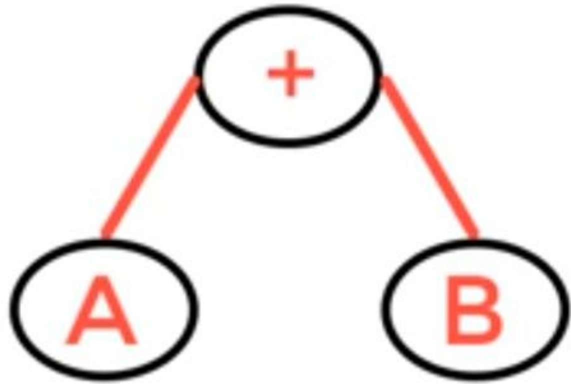
c) current = NULL /*right of 3 */



Traversal is done now as stack S is empty and current is NULL.

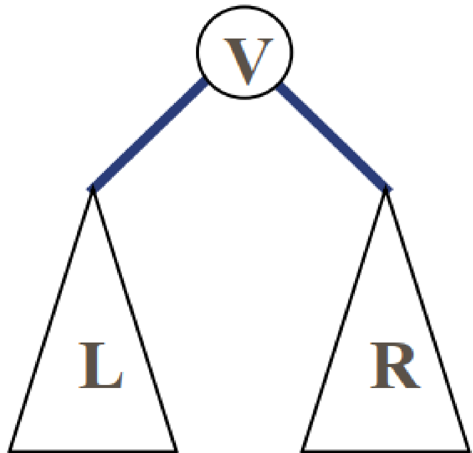
التنقل بأسلوب Pre-order

- يبدأ التنقل بزيارة العقدة (V) ثم الابن الايسر (L) ثم الابن الأيمن (R).
- نرمز له بـ (VLR) أو (root left right).
- مثال: قم بطباعة عقد الشجرة التالية لدى زيارتها بأسلوب pre-order



Pre-order : root left right

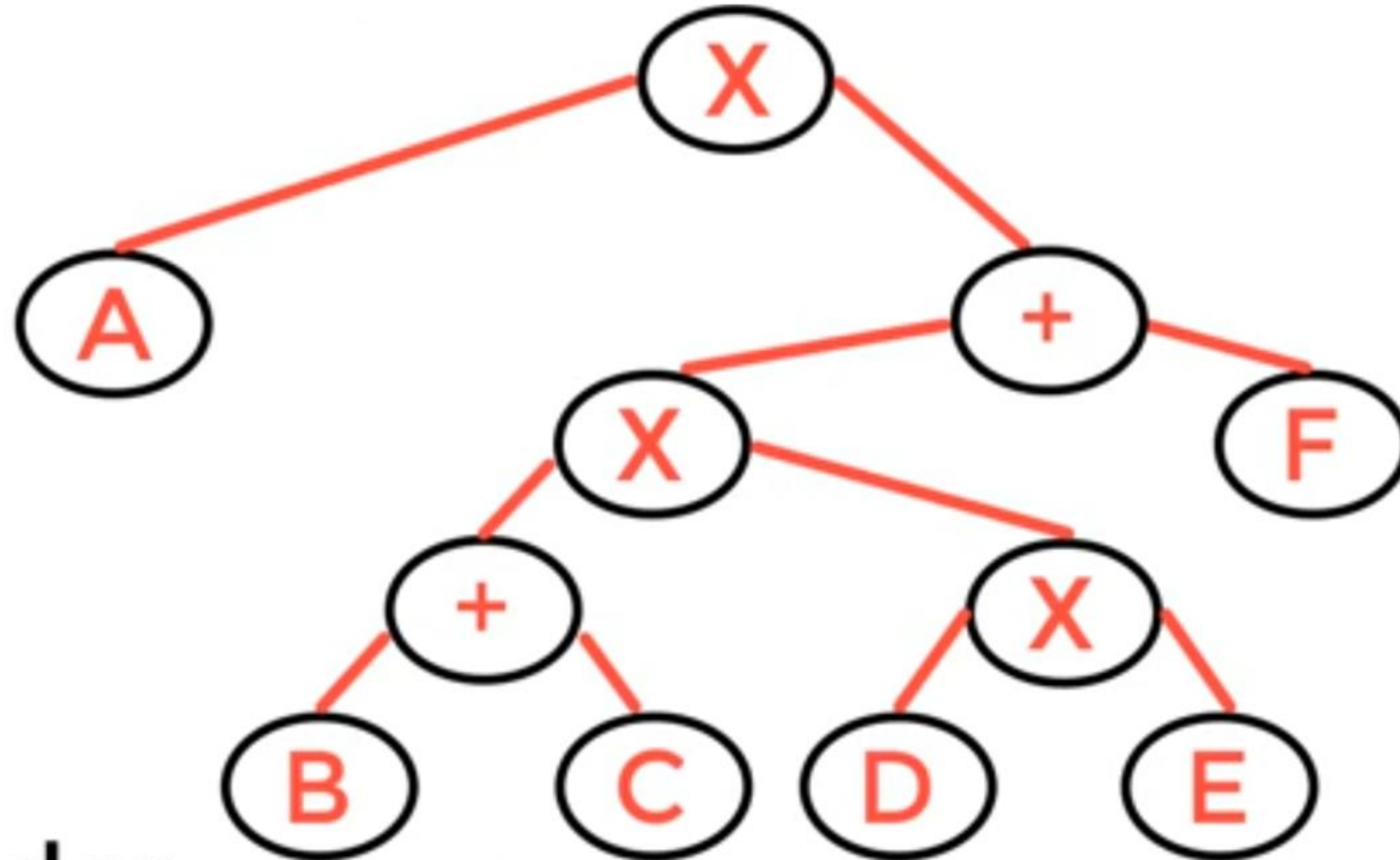
+ A B



- عملياً يتم زيارة العقدة الاب (V) ثم جميع العقد في الشجرة الفرعية اليسارية (Left sub-tree) بأسلوب pre-order ثم جميع العقد في الشجرة الفرعية اليمينية (Right sub-tree) بأسلوب pre-order.

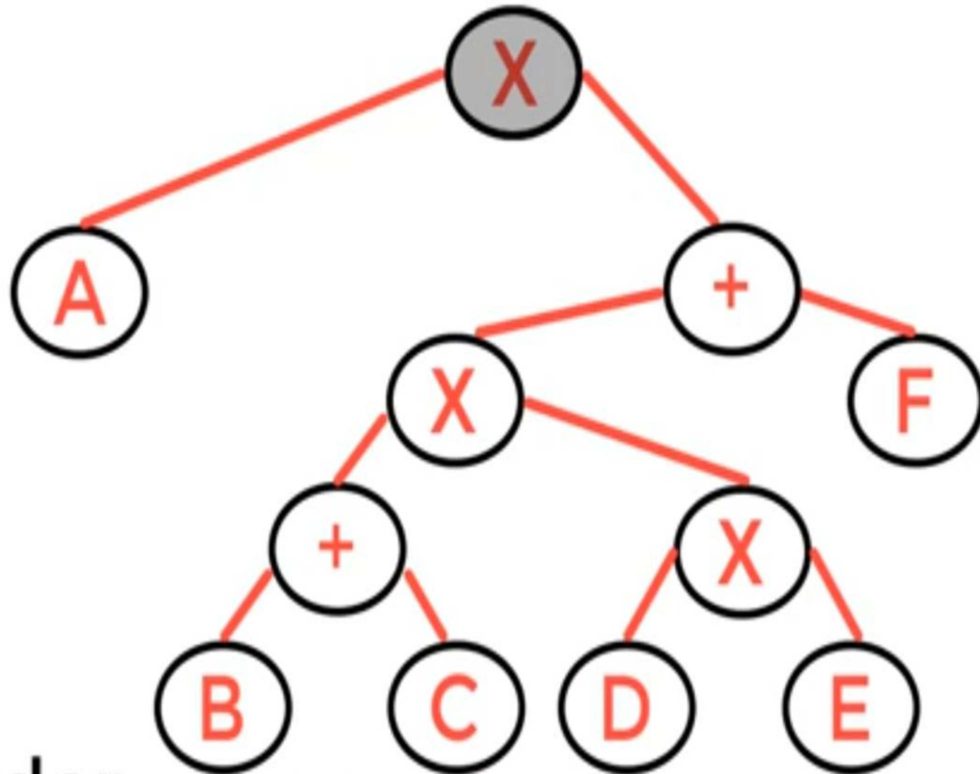
التنقل بأسلوب Pre-order

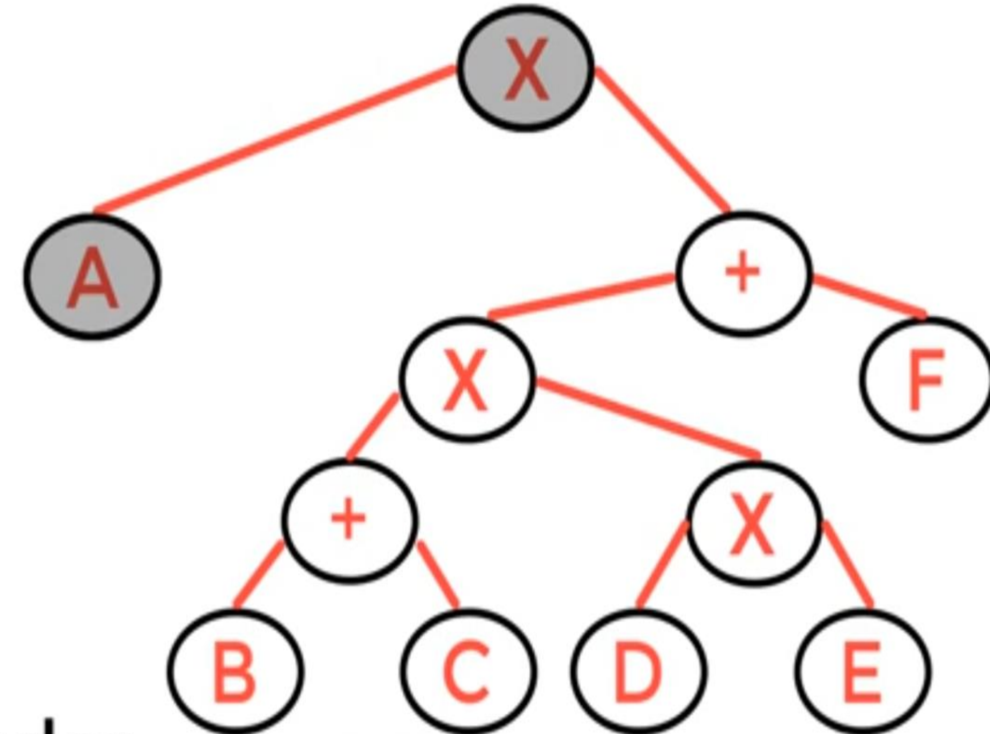
- مثال:



التنقل بأسلوب Pre-order

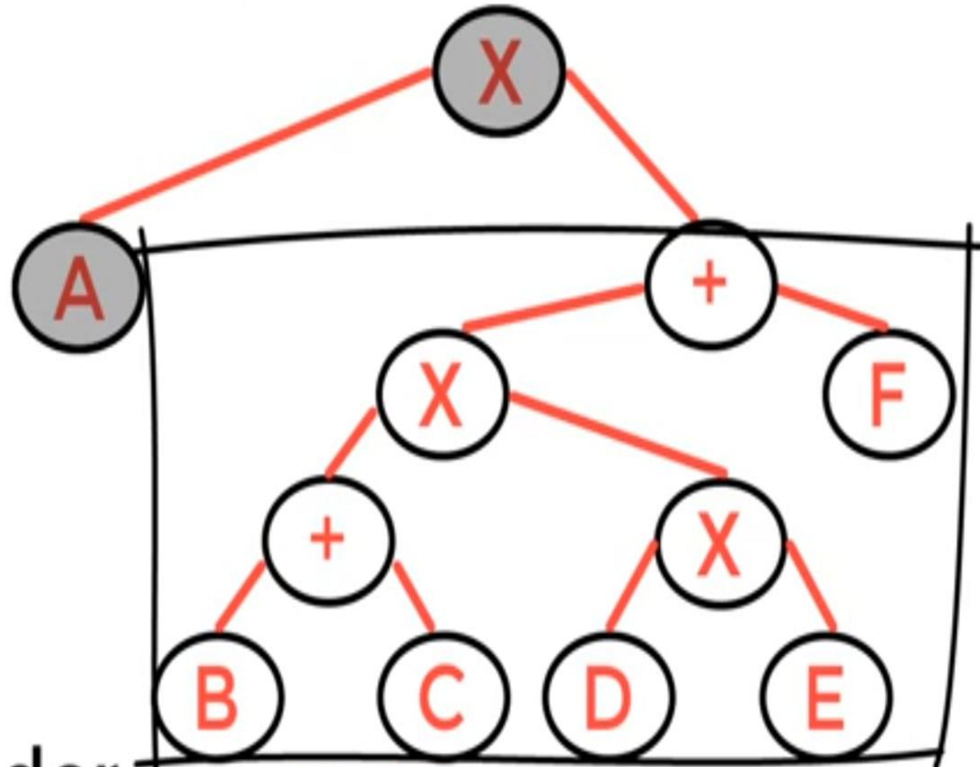
- مثال:

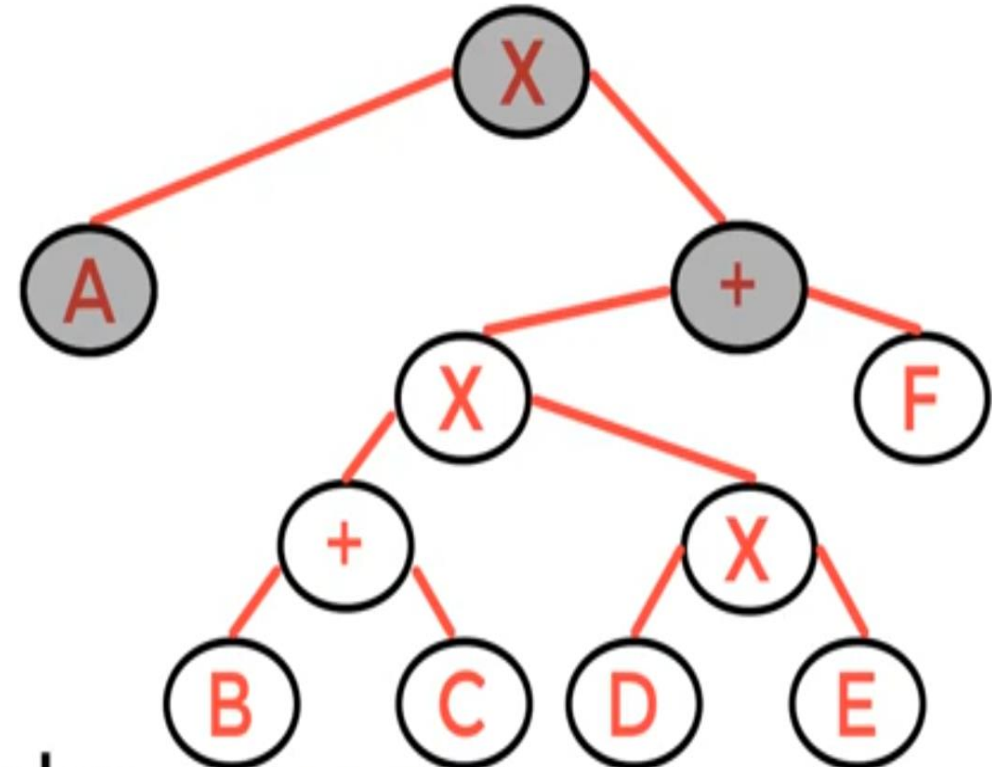




التنقل بأسلوب Pre–order

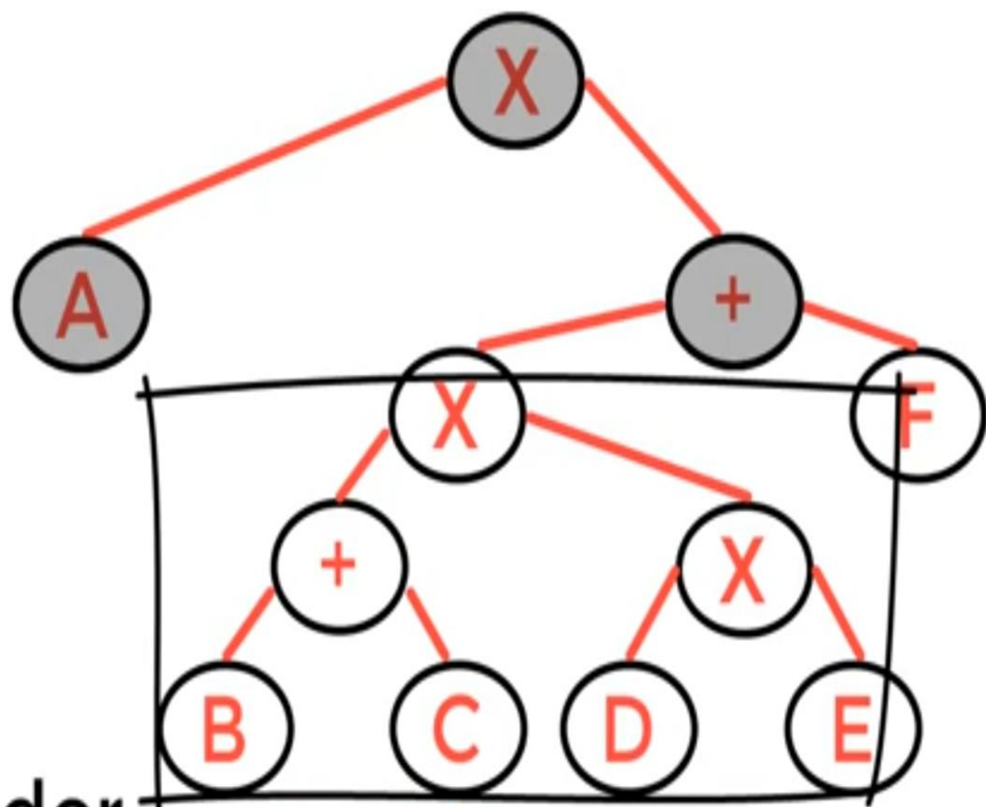
- مثال:

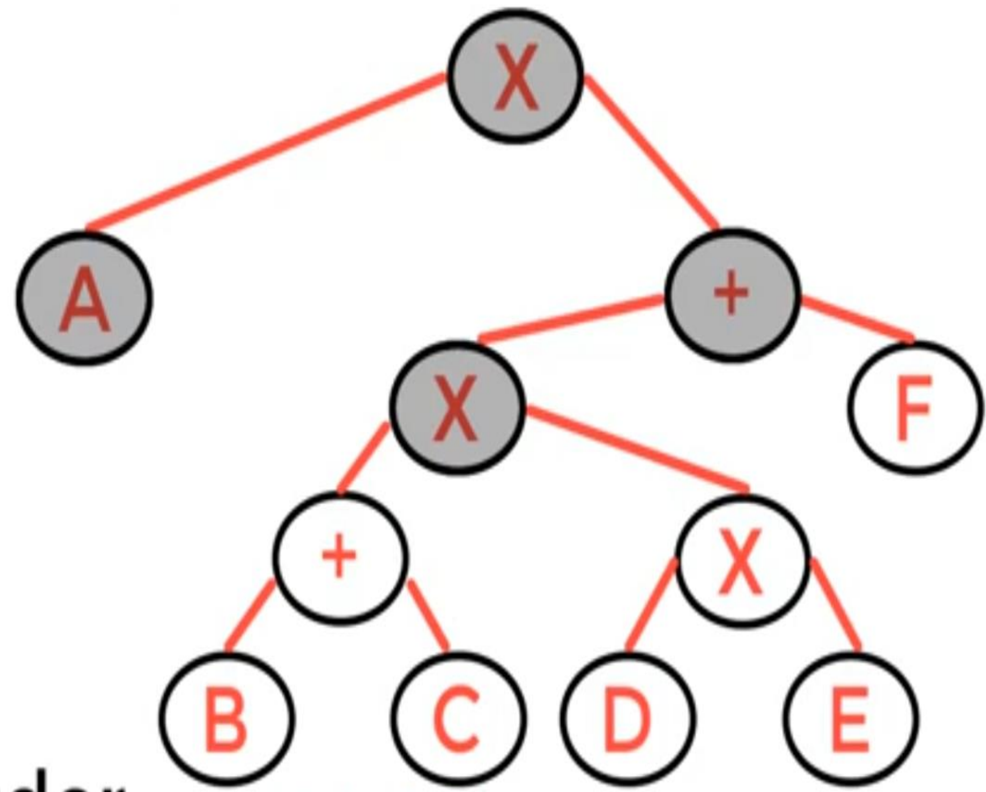




التنقل بأسلوب Pre-order

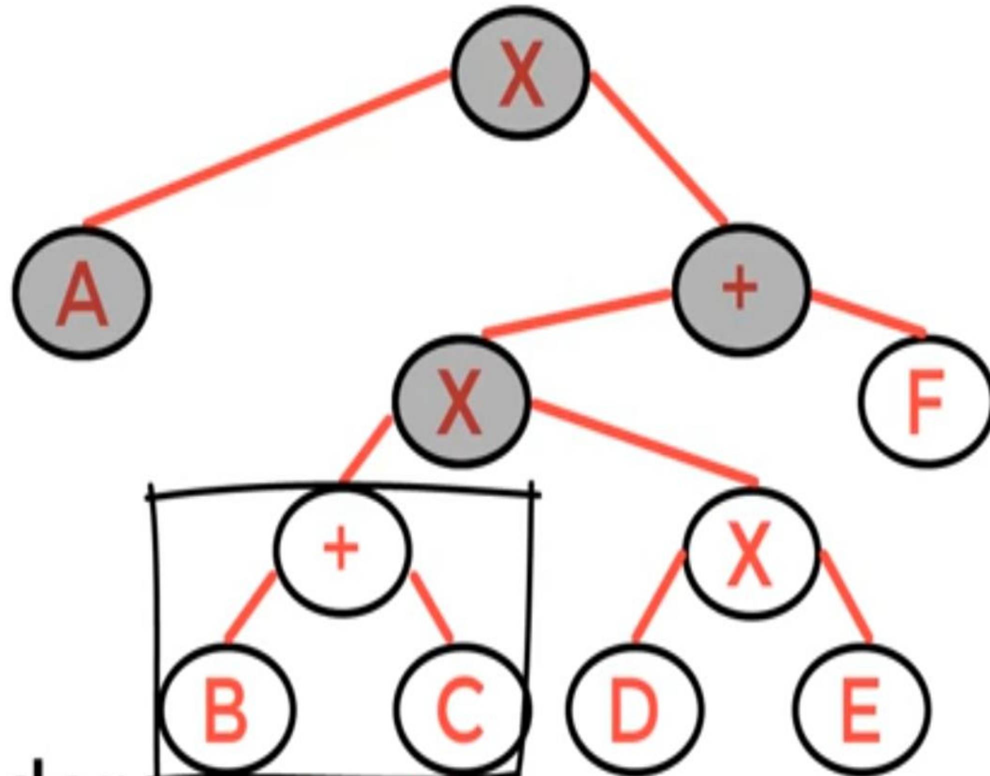
- مثال:

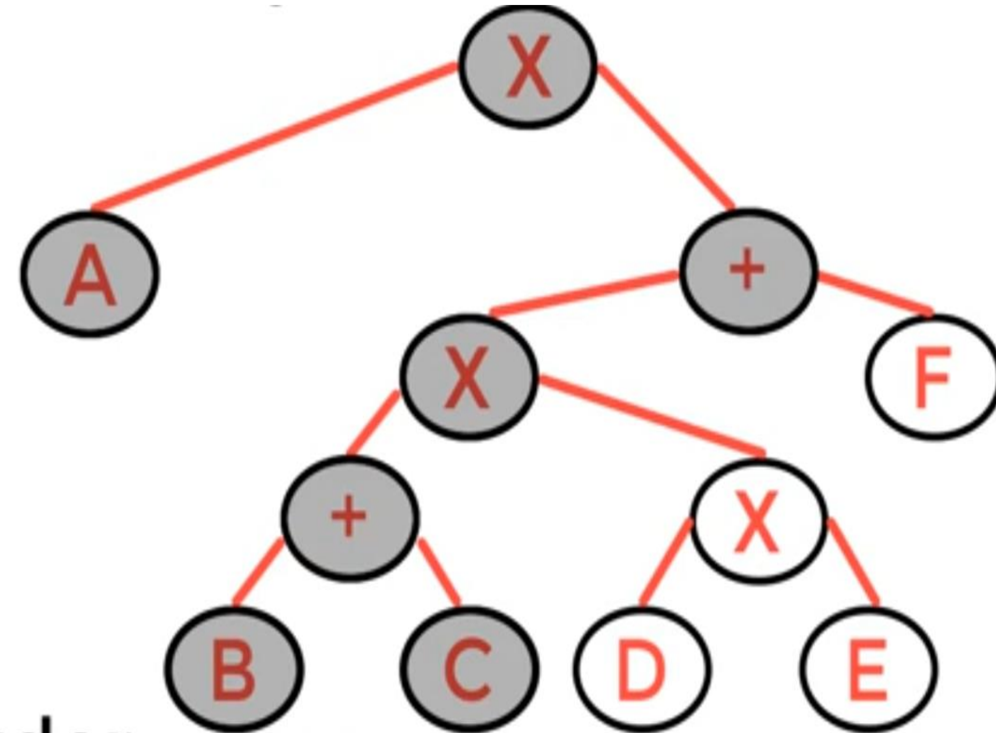




التنقل بأسلوب Pre–order

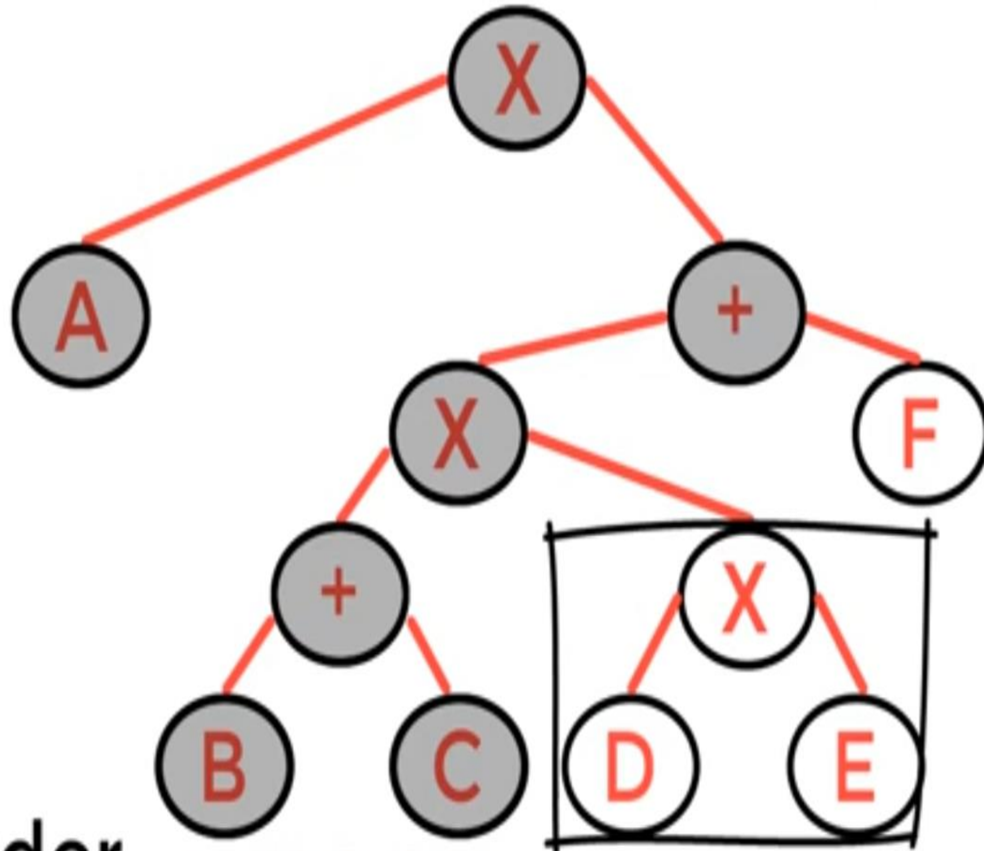
- مثال:

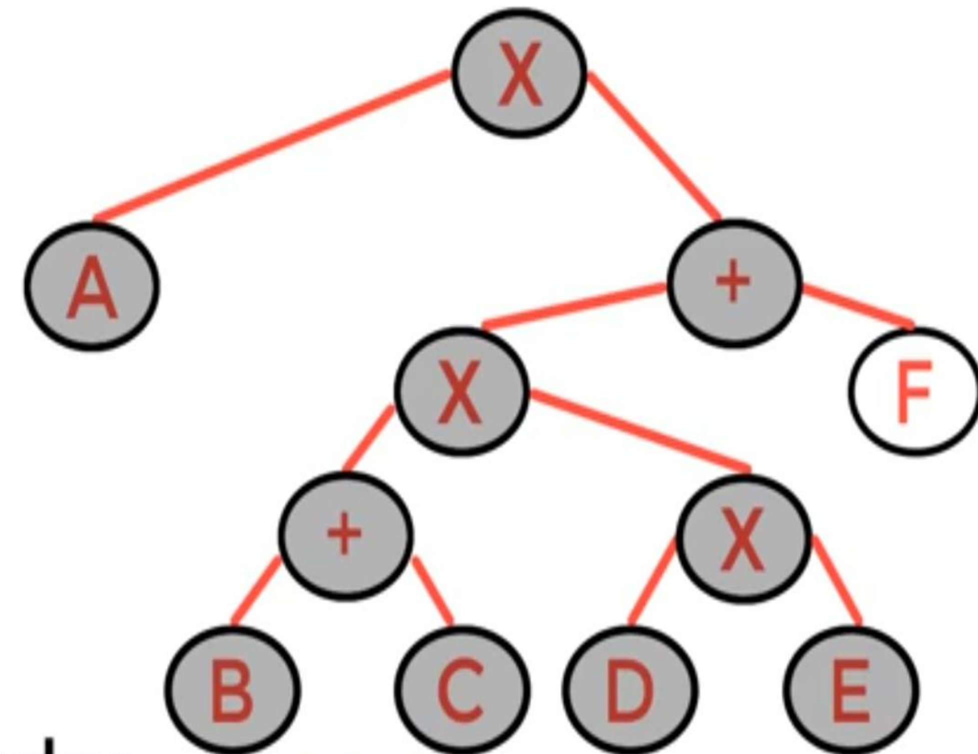




التنقل بأسلوب Pre-order

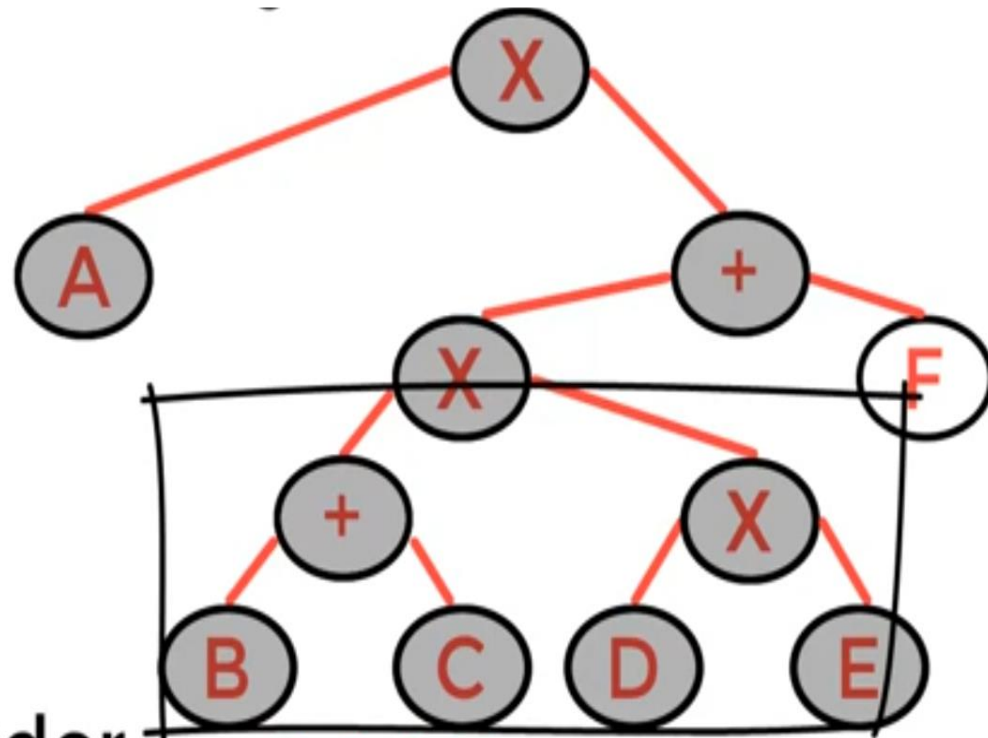
- مثال:

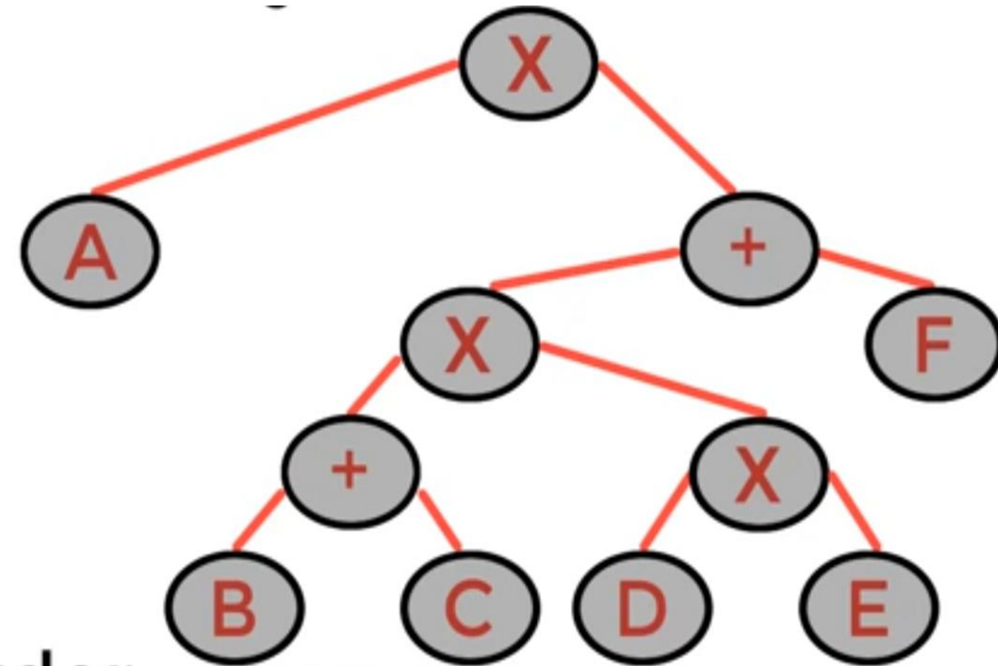




التنقل بأسلوب Pre-order

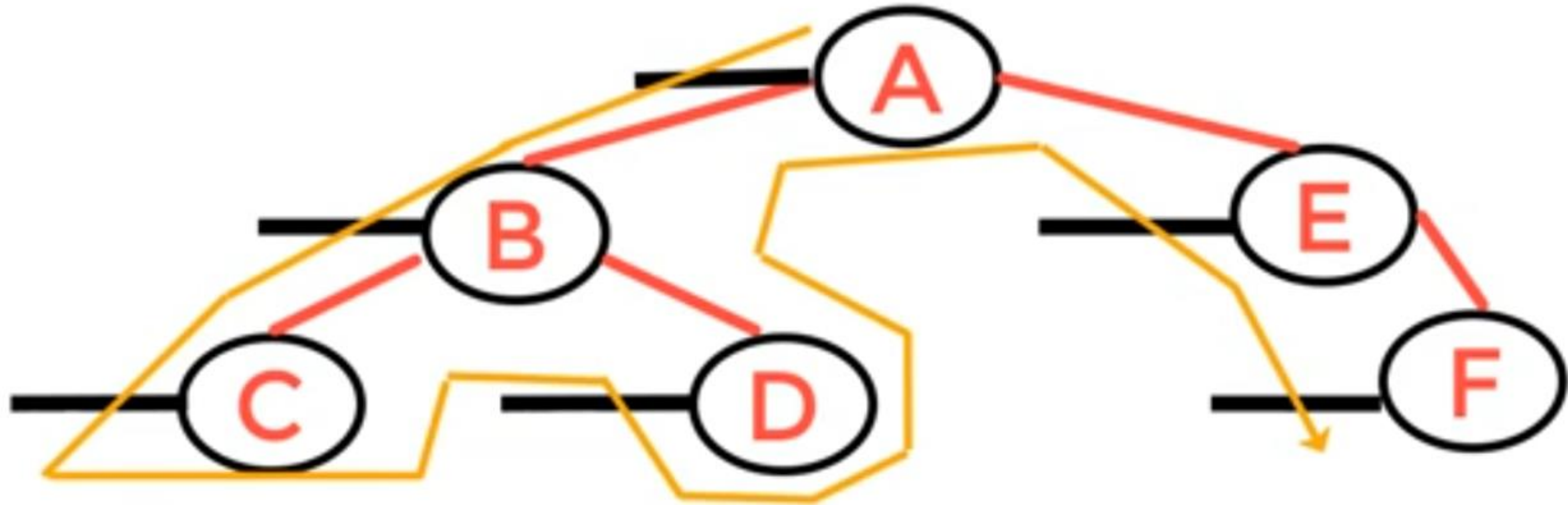
- مثال:





التنقل بأسلوب Pre–order

- مثال:



Pre-order : root left right
A B C D E F

التنقل بأسلوب Pre-order

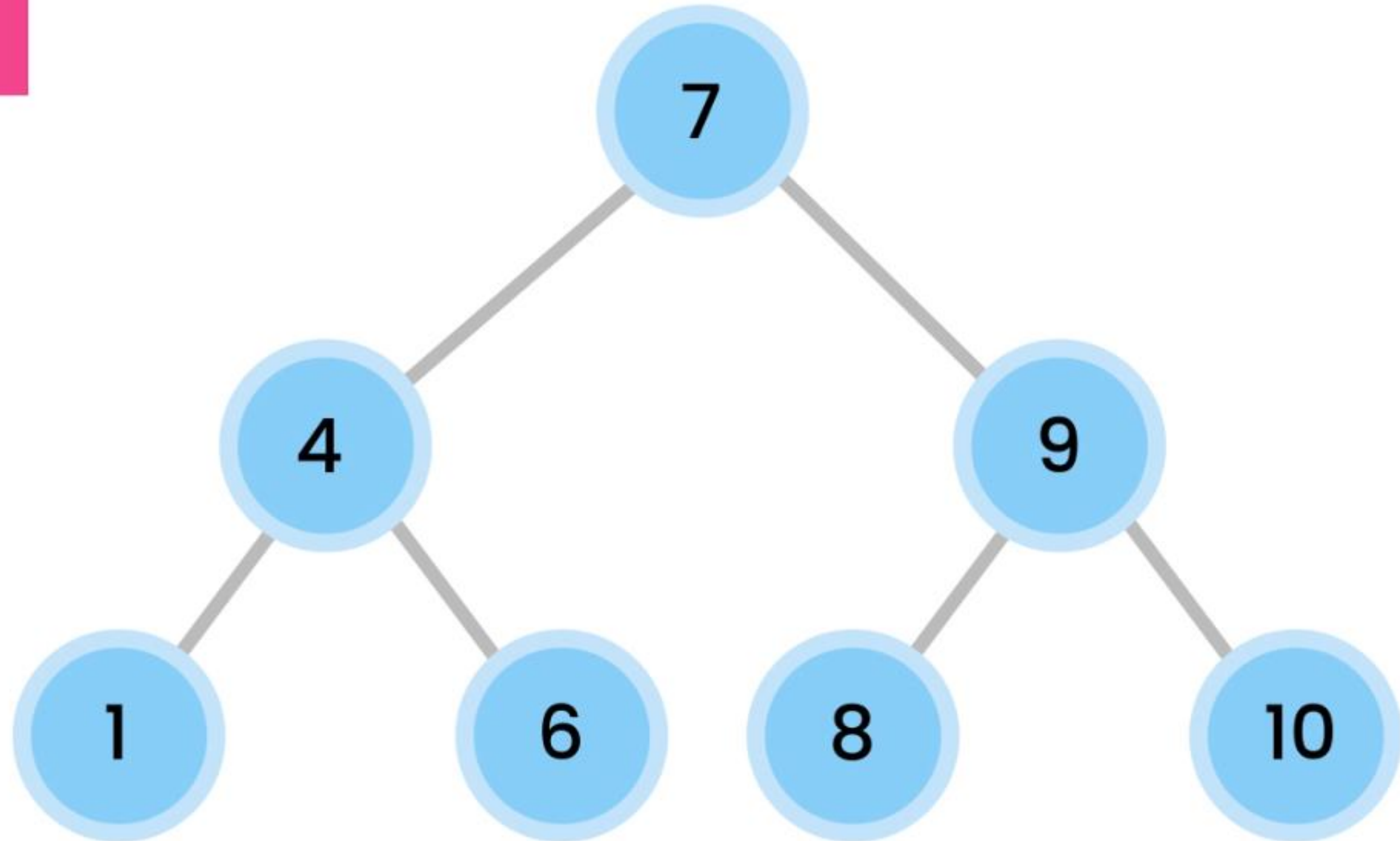
- تعطى الدالة المسؤولة عن تنفيذ هذا التنقل بالصيغة العودية بالشكل التالي :

```
void preOrder(node *p)
        {
         if (p!=Null){
            cout<< p->item<<" ";
            preOrder(p->left);
            preOrder(p->right);
                      }
            }
```

left	item	right

التنقل بأسلوب Pre-order

- مثال:



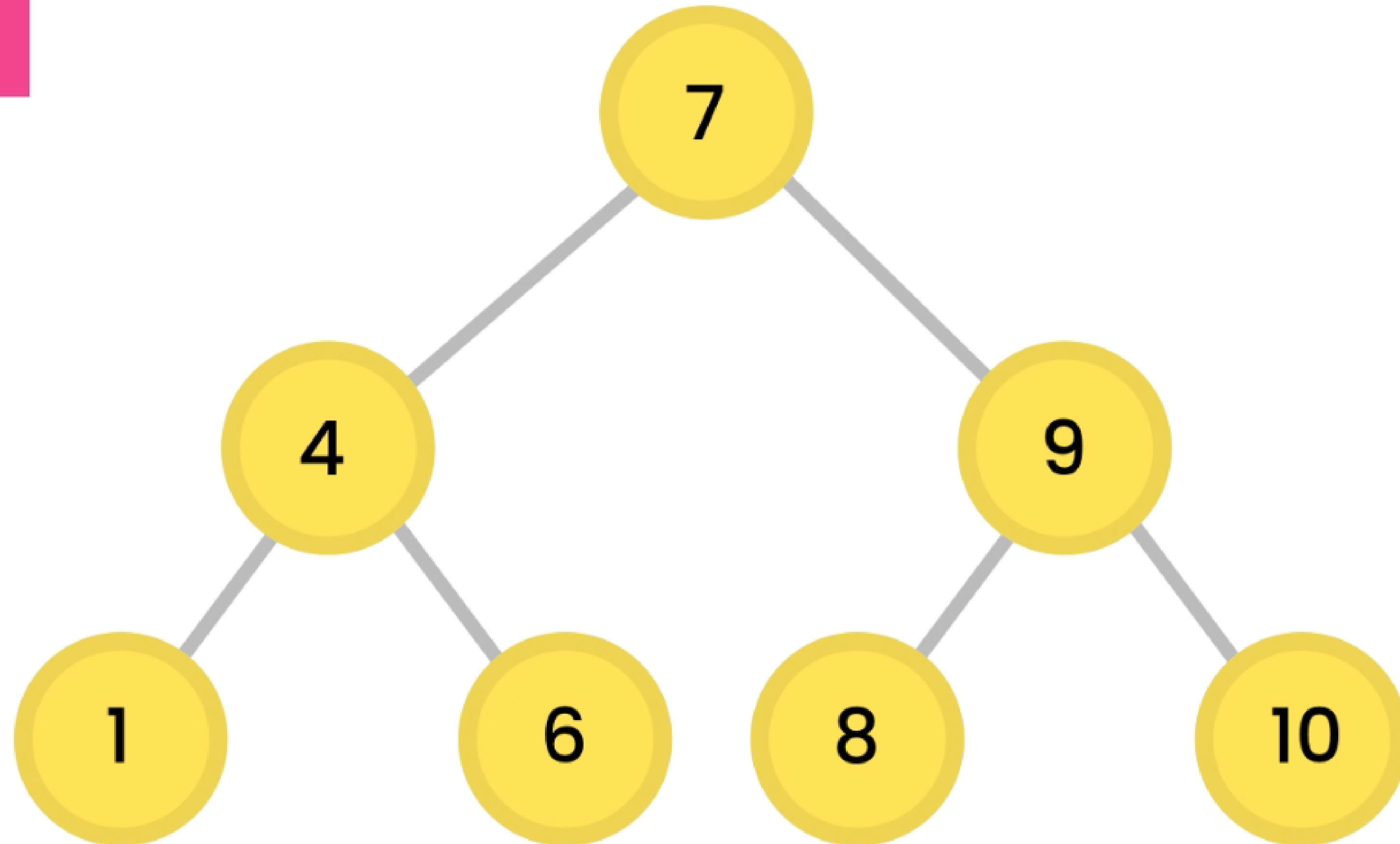
التنقل بأسلوب Pre-order

- مثال:

PRE-ORDER

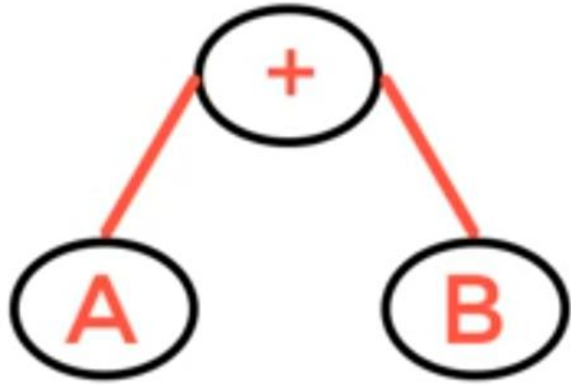
Root, Left, Right

7, 4, 1, 6, 9, 8, 10



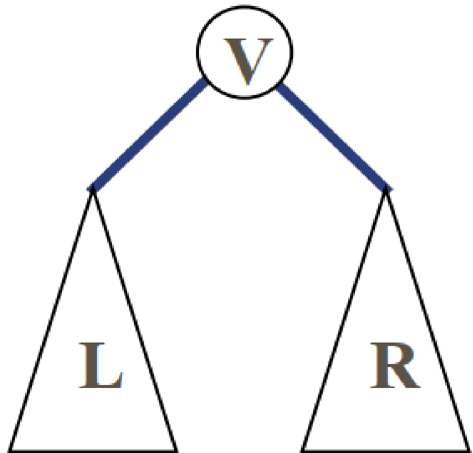
التنقل بأسلوب post–order

- يبدأ التنقل بزيادة الابن الايسر (L) ثم الابن الأيمن (R) ثم العقدة (V).
- نرمز له بـ (LRV) أو (left right root).
- مثال: قم بطباعة عقد الشجرة التالية لدى زيارتها بأسلوب post–order



Post-order : left right root

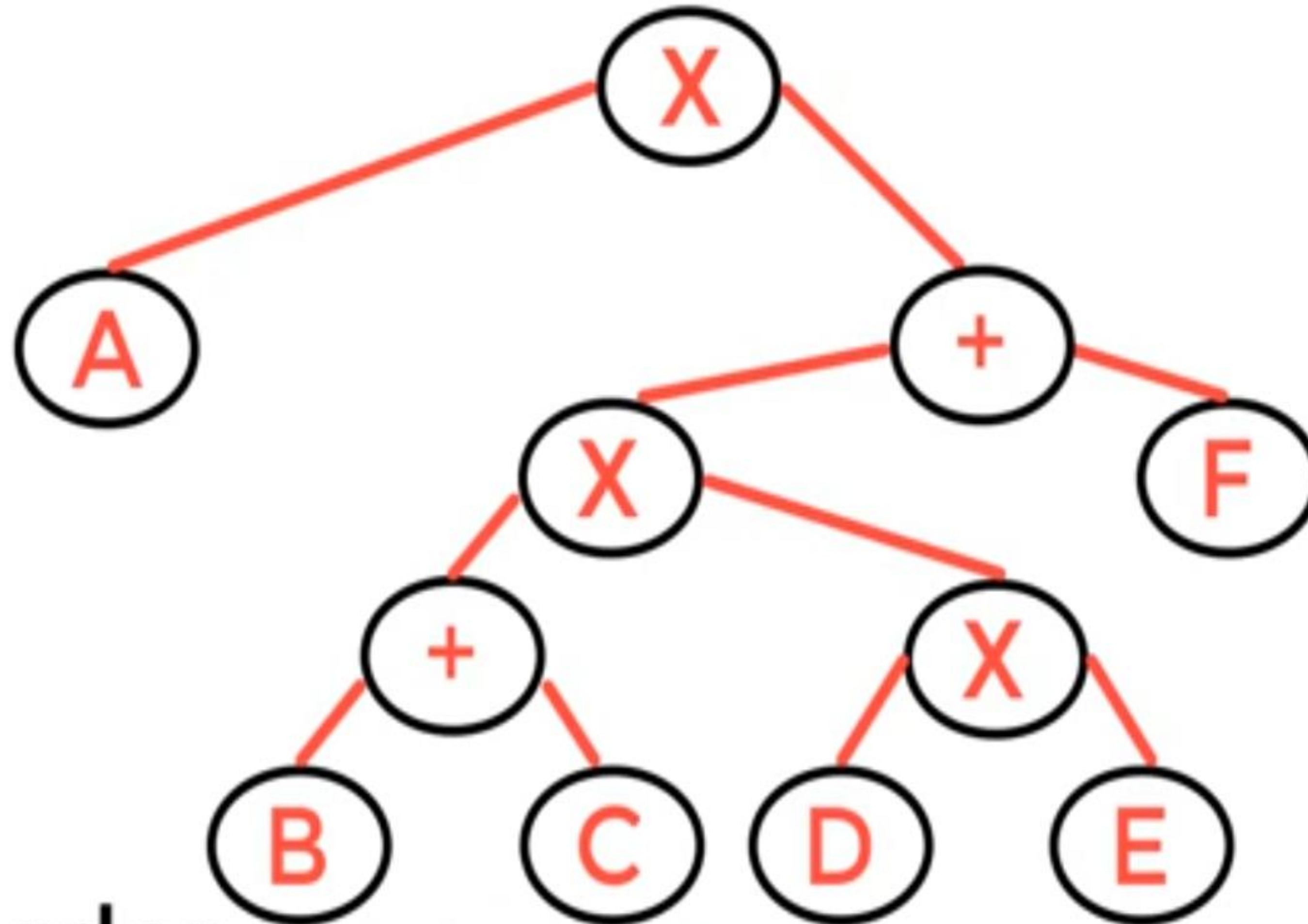
A B +



- عملياً يتم زيارة جميع العقد في الشجرة الفرعية اليسارية (Left sub–tree) بأسلوب post–order ثم جميع العقد في الشجرة الفرعية اليمينية (Right sub–tree) بأسلوب post–order ثم العقدة الاب (V).

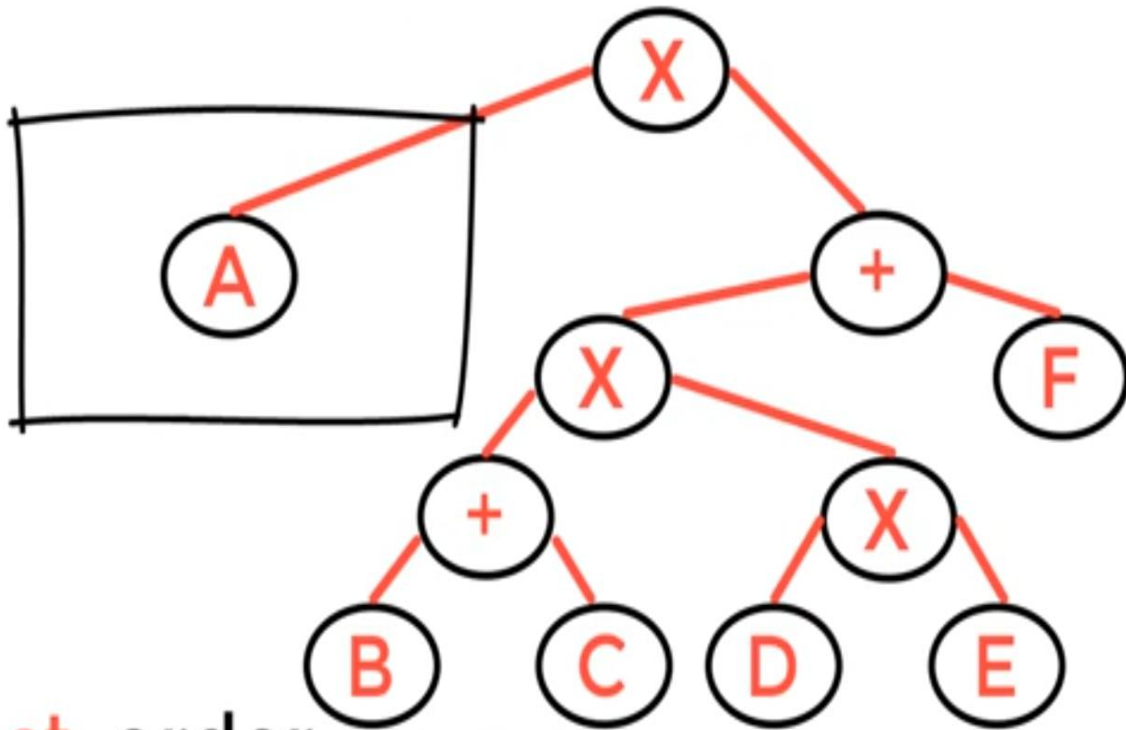
التنقل بأسلوب post–order

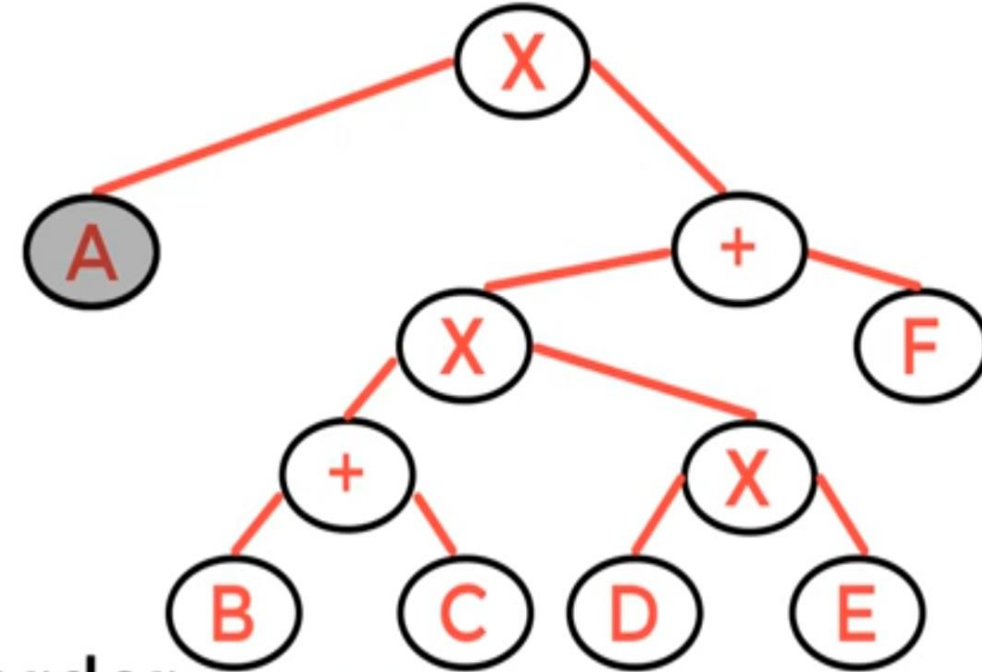
- مثال:



التنقل بأسلوب post-order

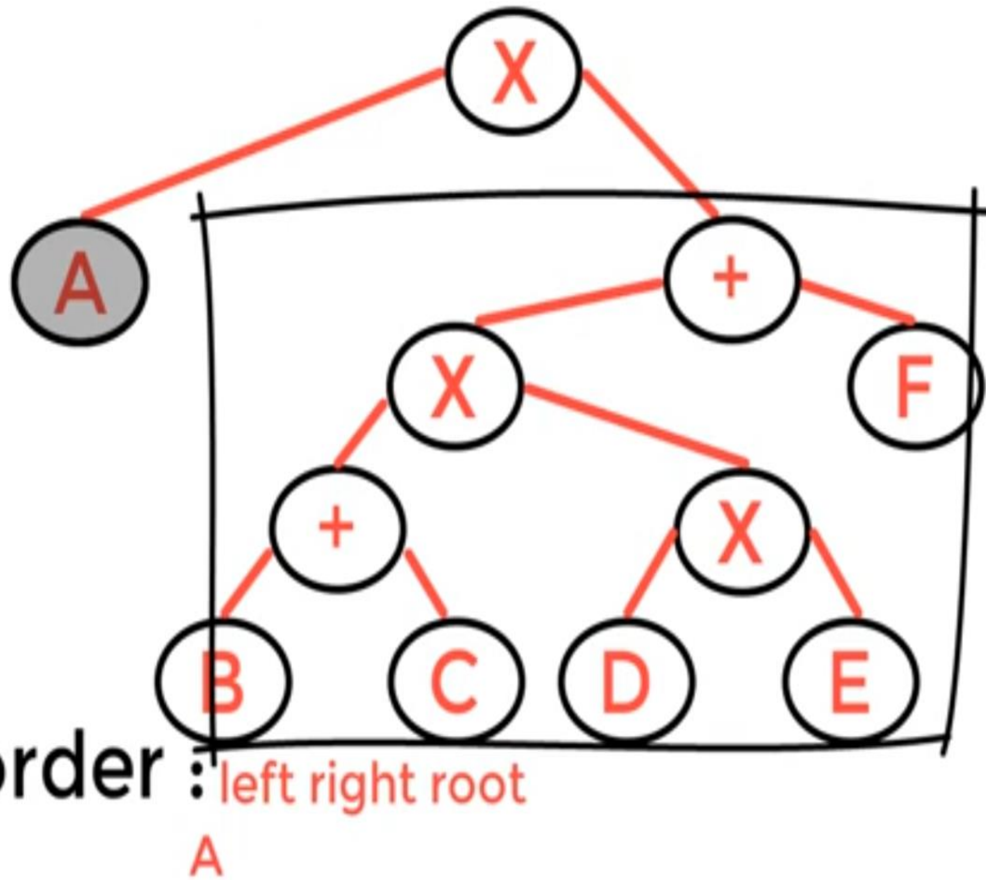
- مثال:

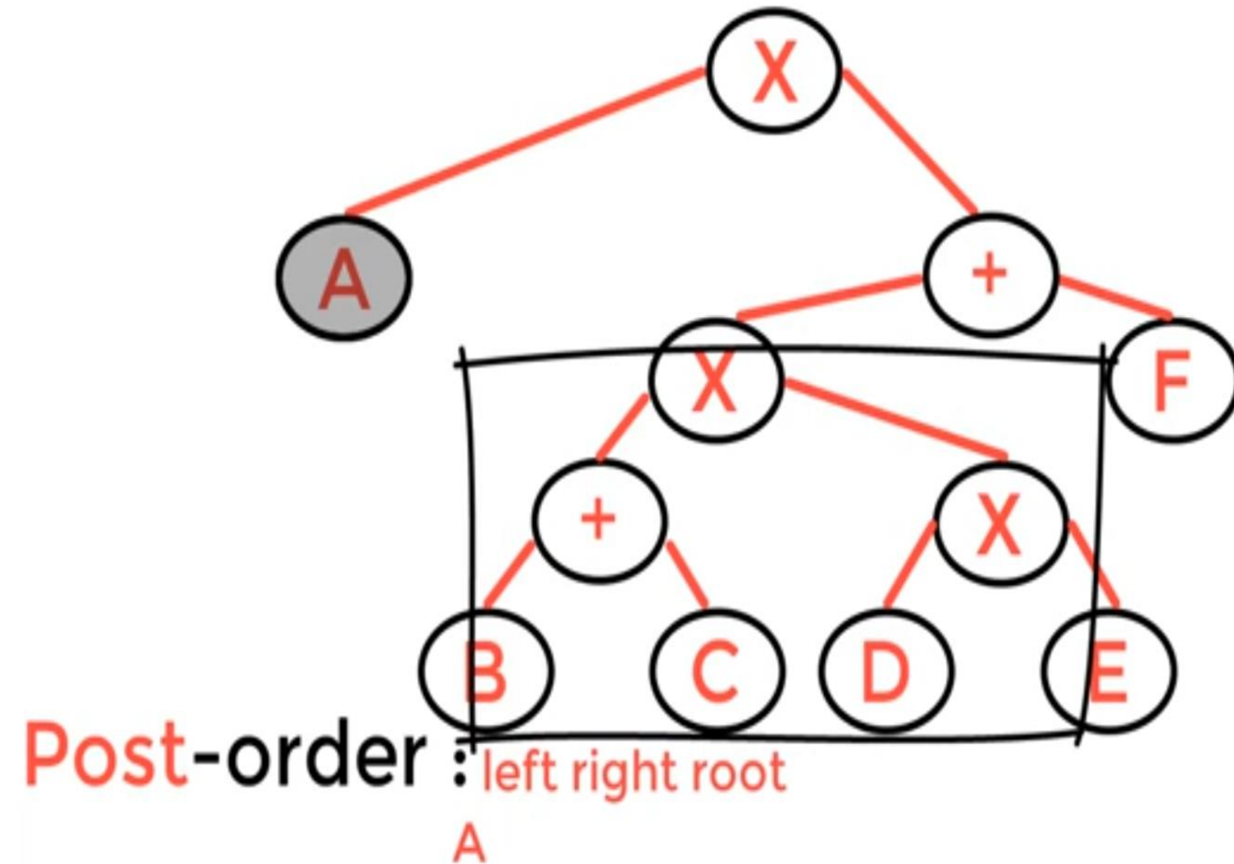




التنقل بأسلوب post–order

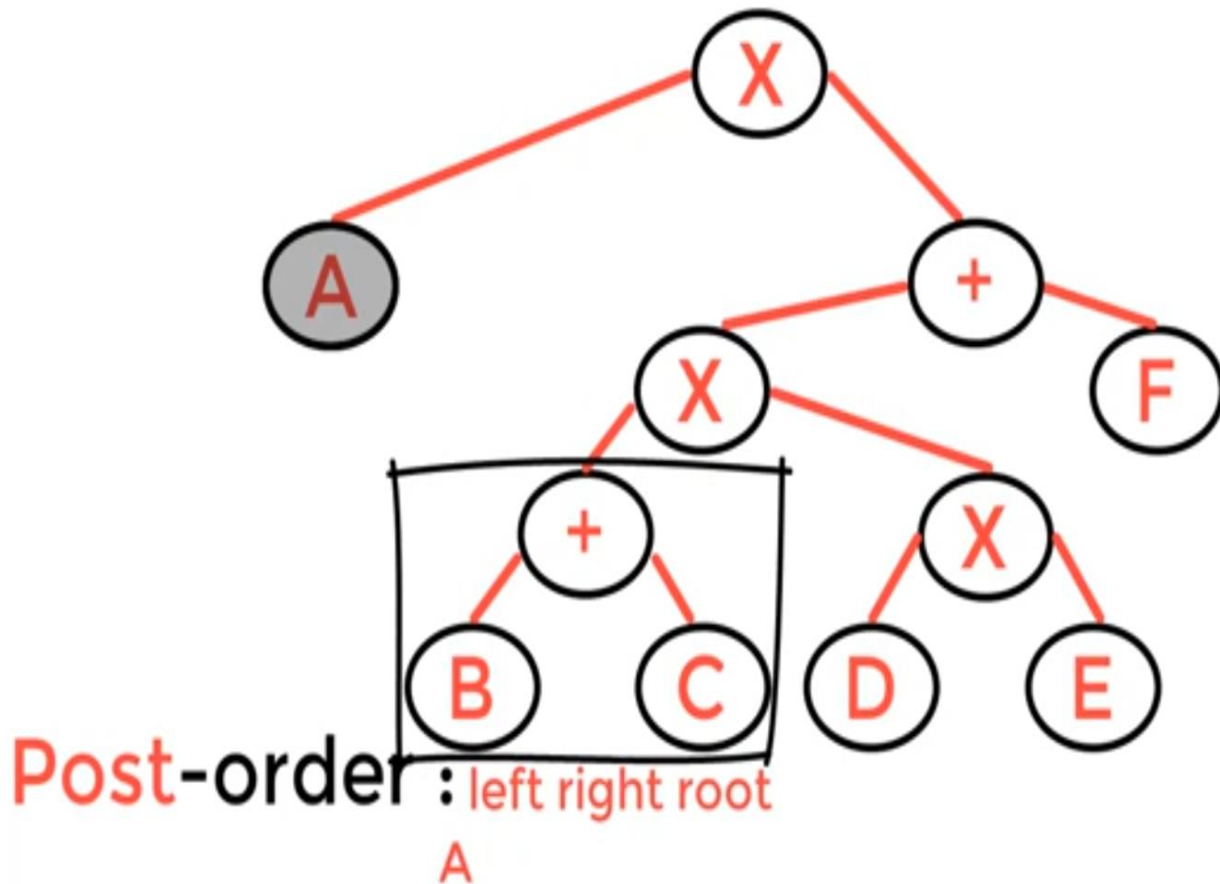
- مثال:

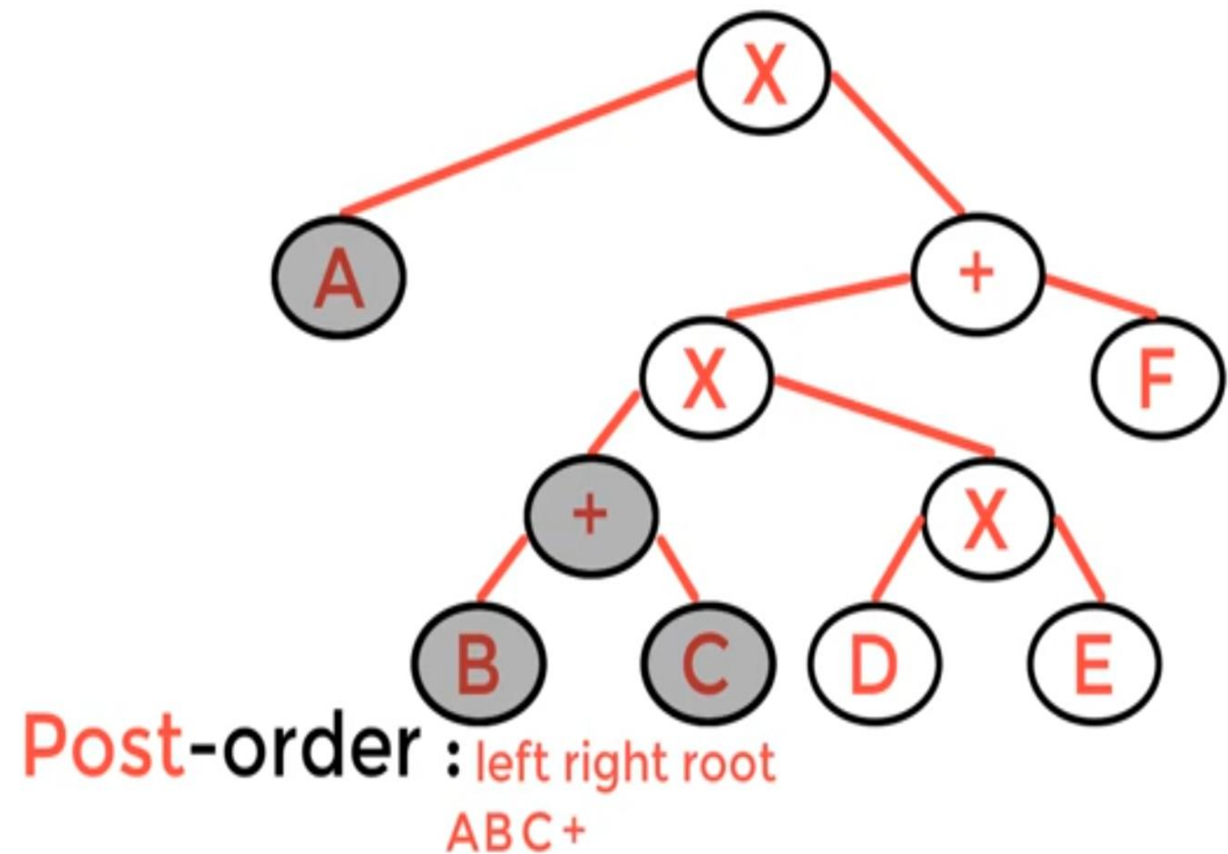




التنقل بأسلوب post–order

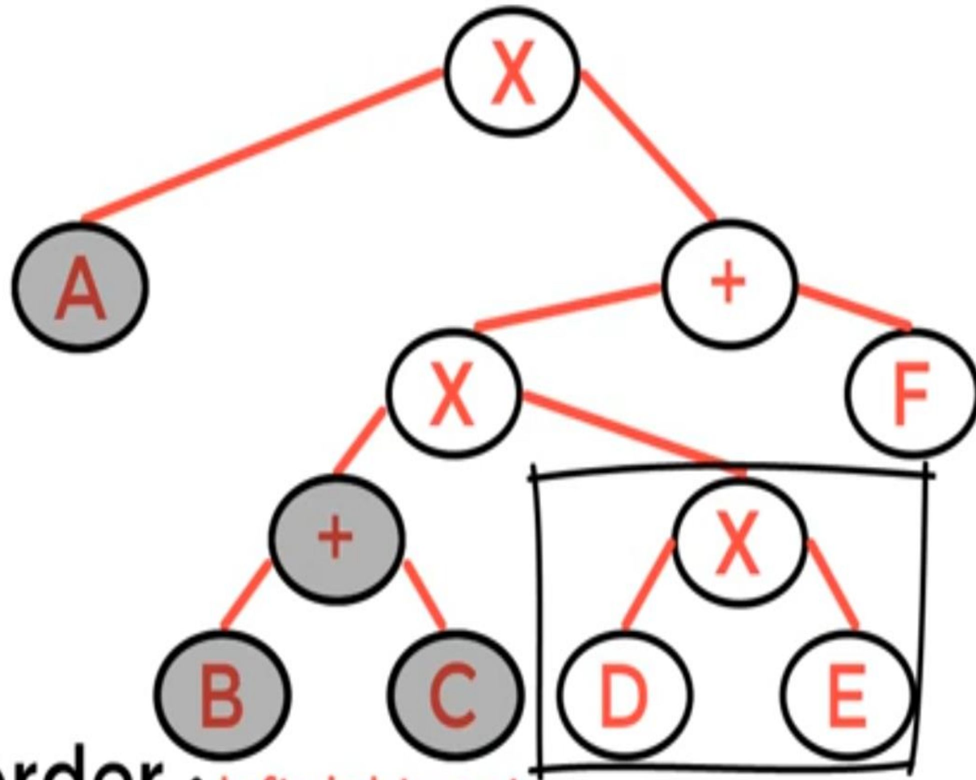
- مثال:

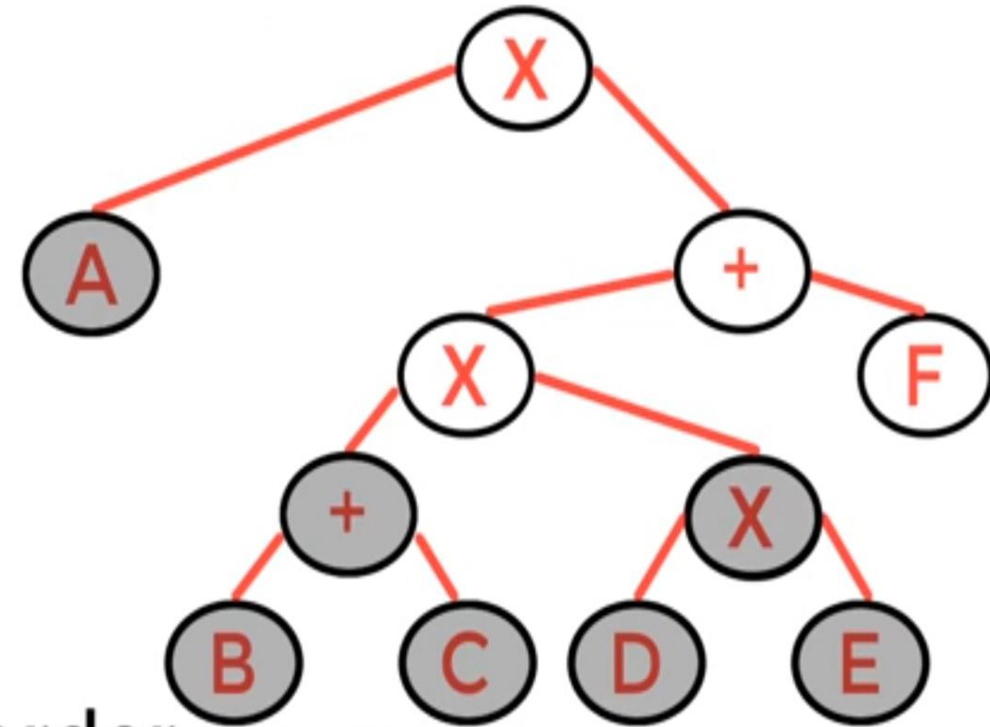




التنقل بأسلوب post–order

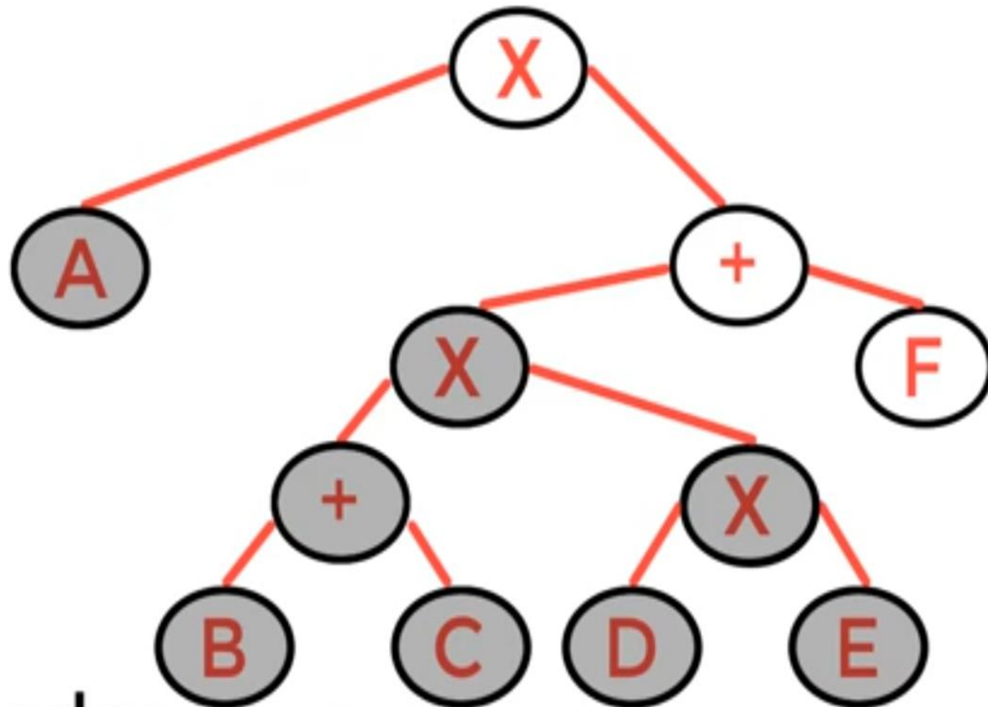
- مثال:

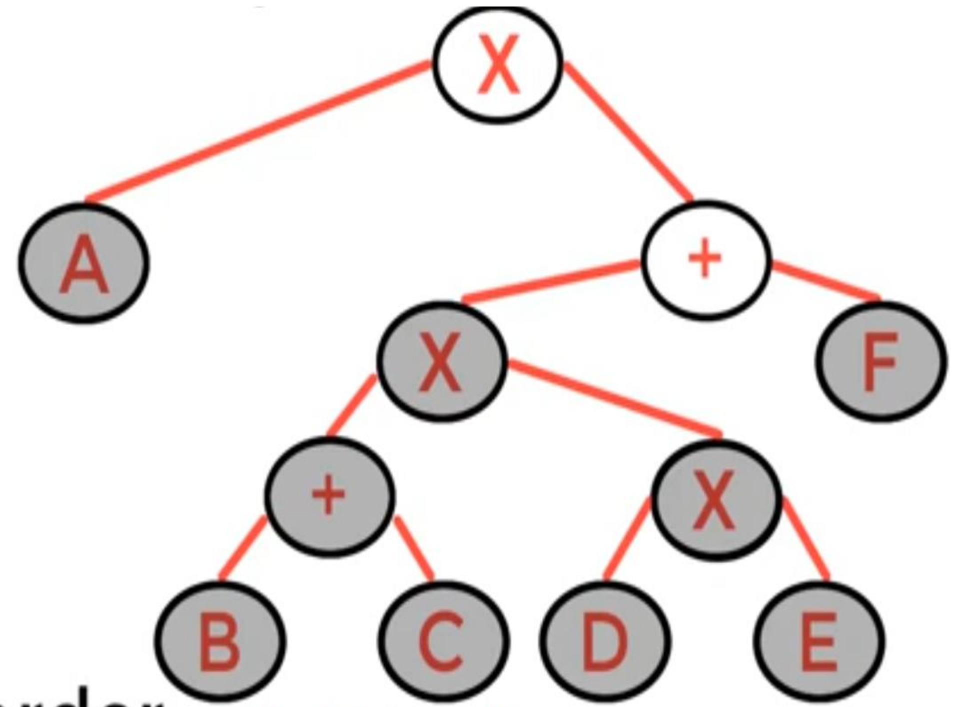




التنقل بأسلوب post–order

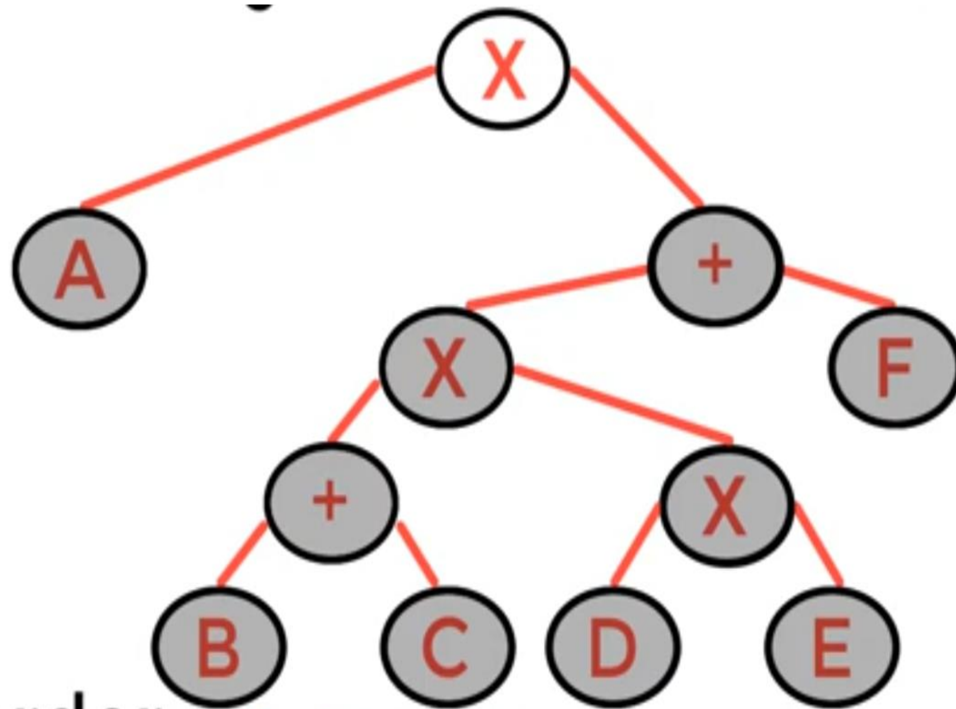
- مثال:

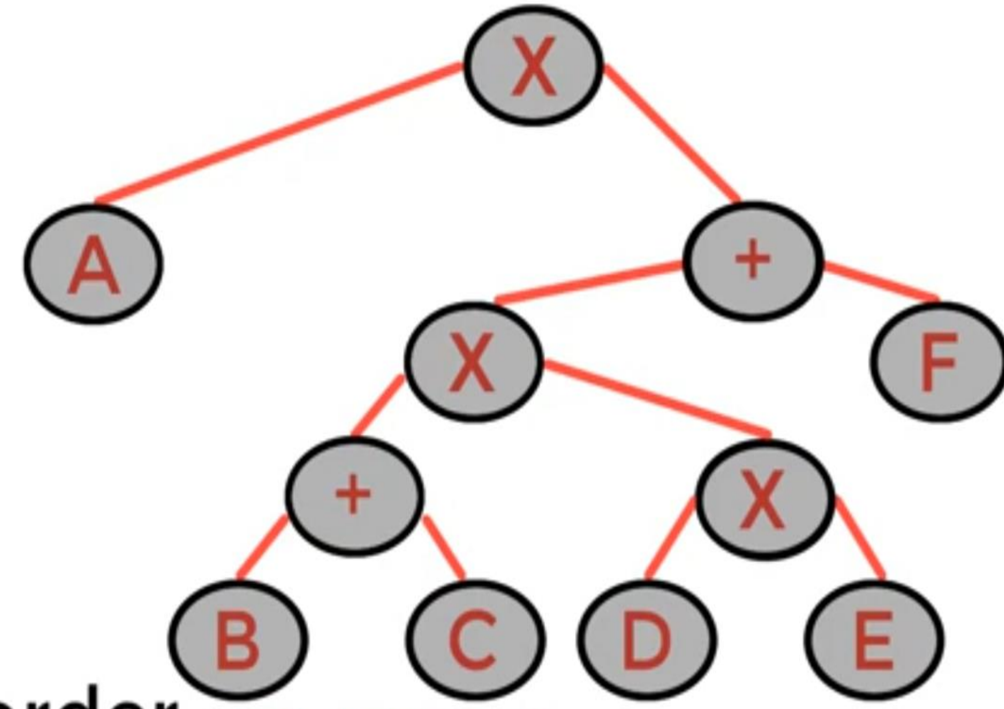




التنقل بأسلوب post–order

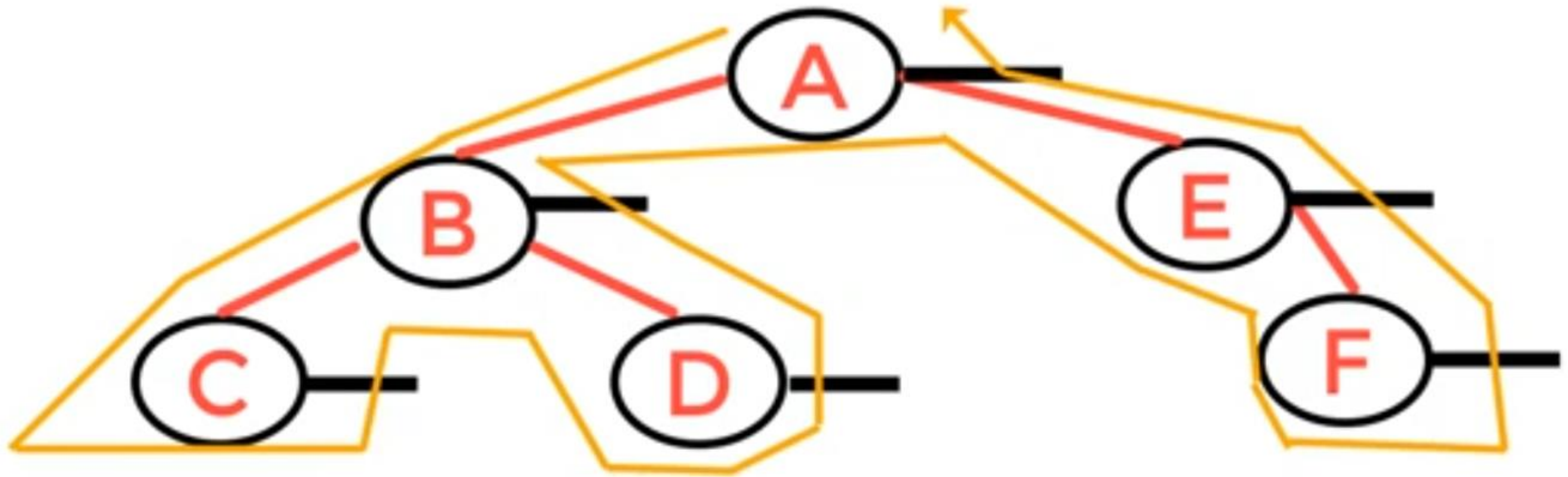
- مثال:





التنقل بأسلوب post–order

- مثال:



Post-order : left right root

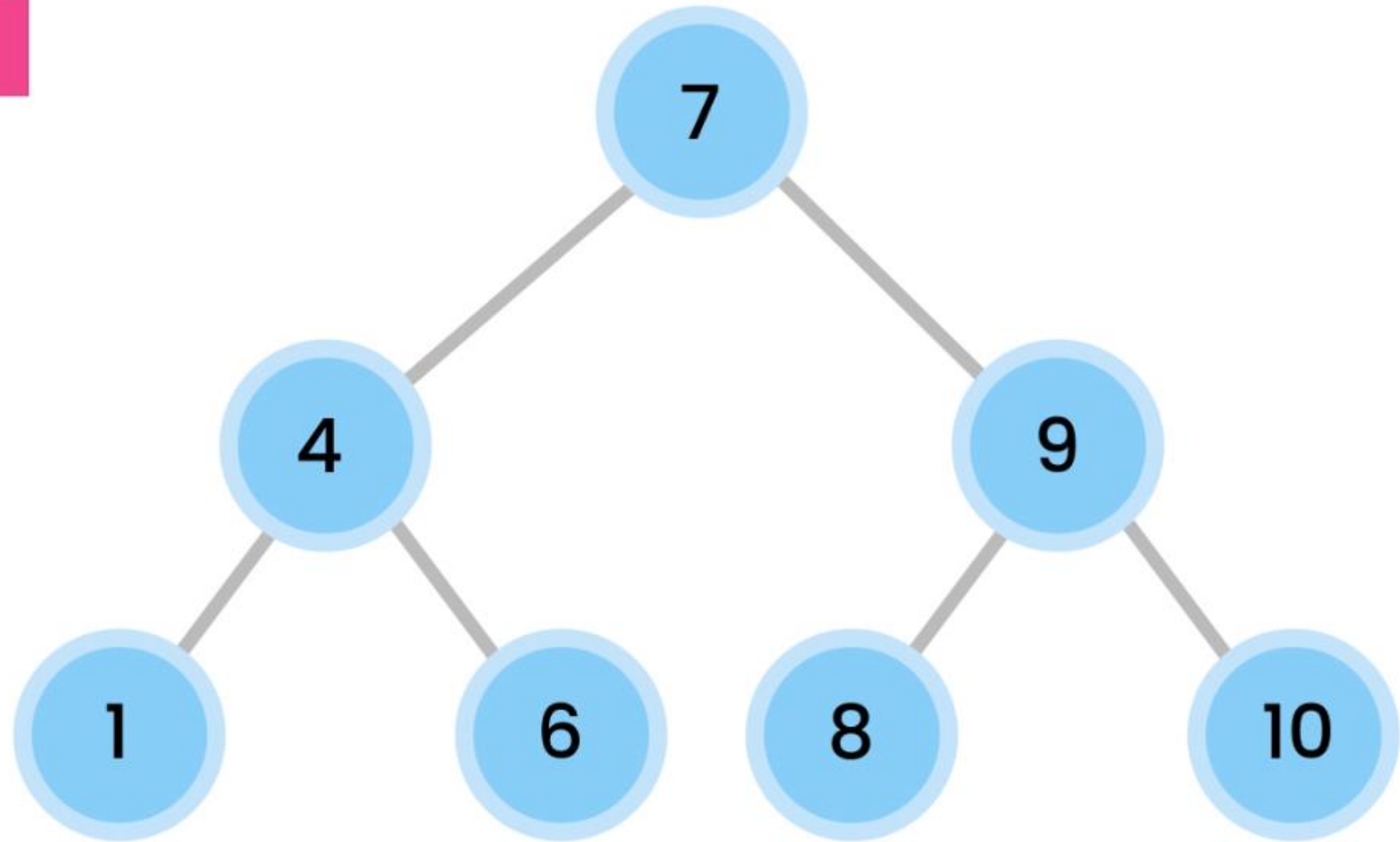
CDBFEA

التنقل بأسلوب post–order

- مثال:

POST-ORDER

Left, Right, Root



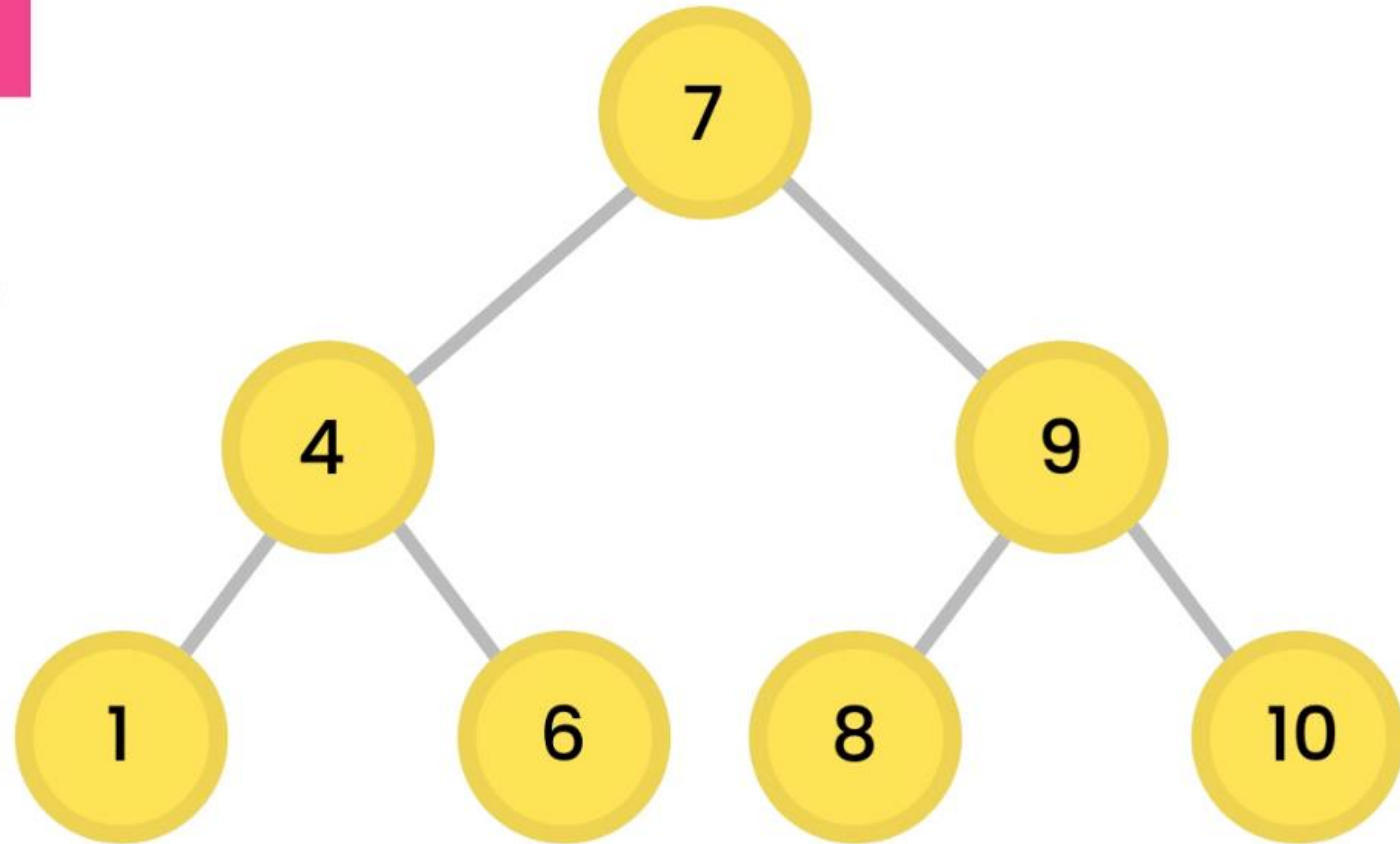
التنقل بأسلوب post–order

- مثال:

POST-ORDER

Left, Right, Root

1, 6, 4, 8, 10, 9, 7



التنقل بأسلوب post-order

- تعطى الدالة المسؤولة عن تنفيذ هذا التنقل بالصيغة العودية بالشكل التالي :

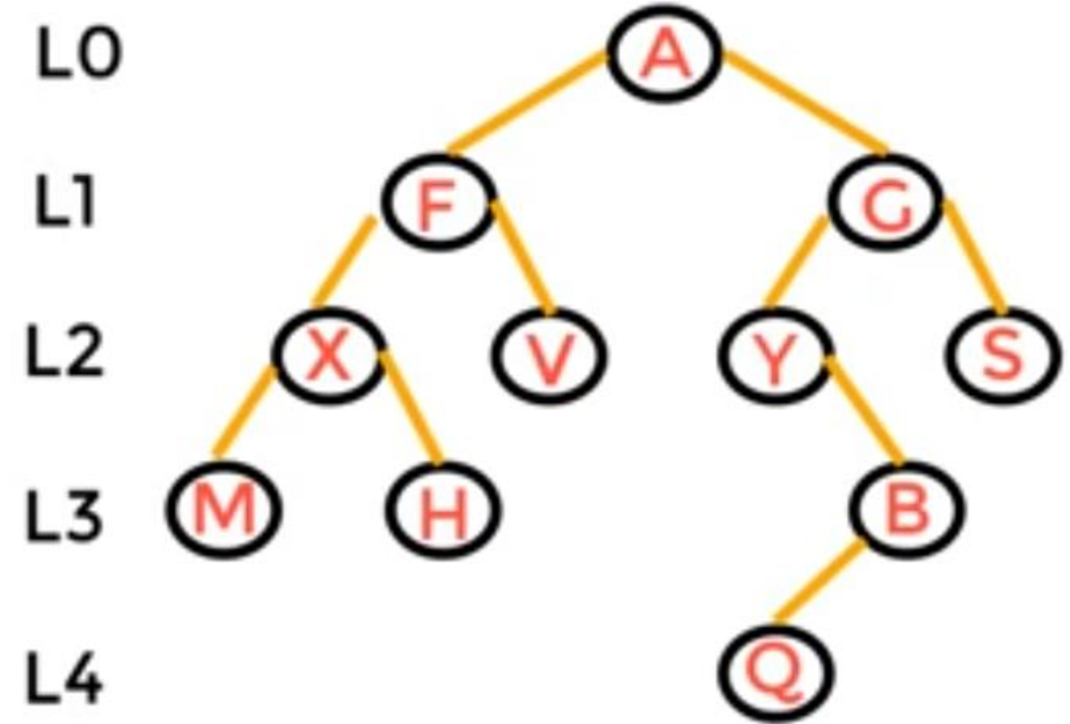
```
void postOrder(node *p)
        {
         if (p!=Null){
            postOrder(p->left);
            postOrder(p->right);
            cout<< p->item<<" ";
                       }
            }
```

left	item	right

التنقل بأسلوب level–order

Depth-first traversal:

-Pre-order :root left right
-In-order :left root right
-Post-order :left right root
:root right left
:right root left
:right left root



Breadth-first traversal:

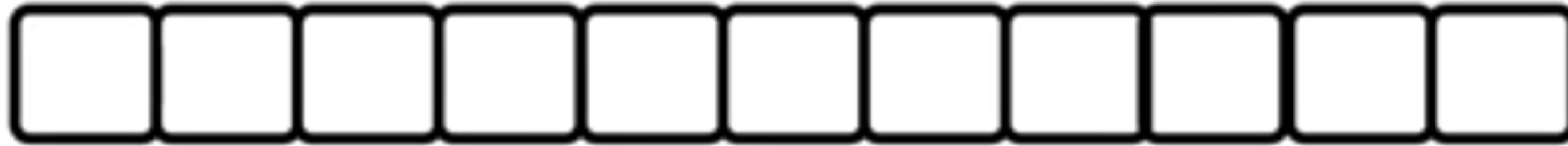
Level-order

A F G X V Y S M H B Q

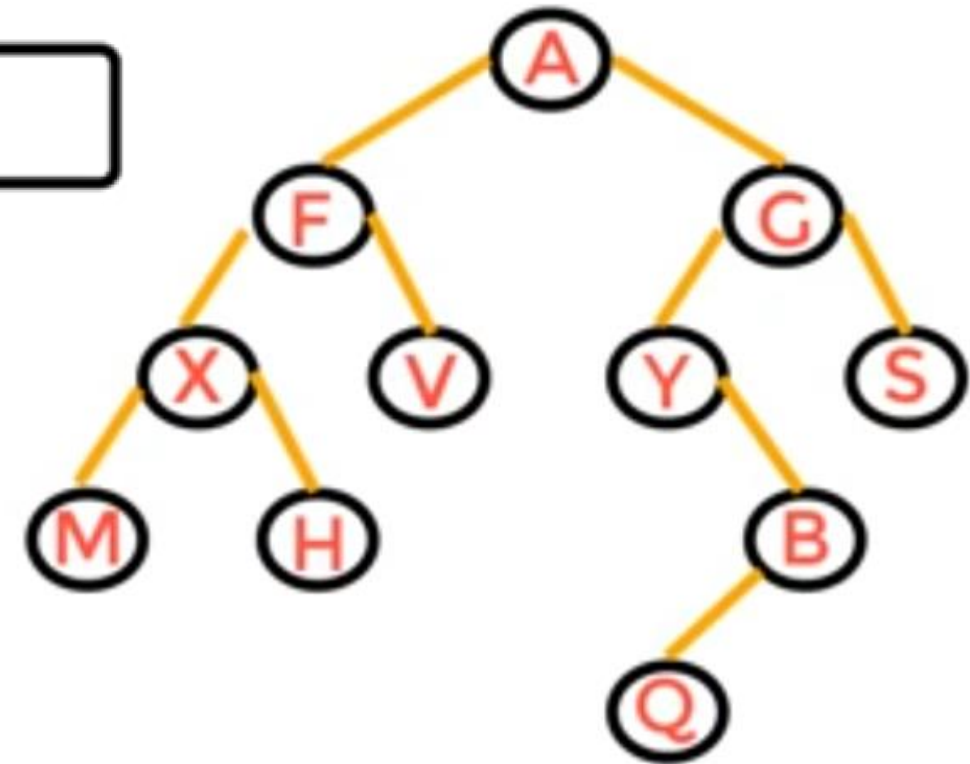
- في هذا الأسلوب من التنقل، يتم زيارة العقد في كل مستوي ابتداء من العقدة الواقعة في أقصى اليسار إلى العقدة الواقعة في أقصى اليمين .

التنقل بأسلوب level–order

- تبدأ الدالة المسؤولة عن تنفيذ هذا التنقل بإضافة عقدة الجذر إلى رتل، ثم يتم و**بشكل تكراري**
  1. حذف عقدة من مقدمة الرتل
  2. طباعة (معالجة) حقل بيانات هذه العقدة،
  3. وبعد ذلك يتم إضافة الابن اليساري ثم الابن اليميني لهذه العقدة إلى الرتل.



Queue

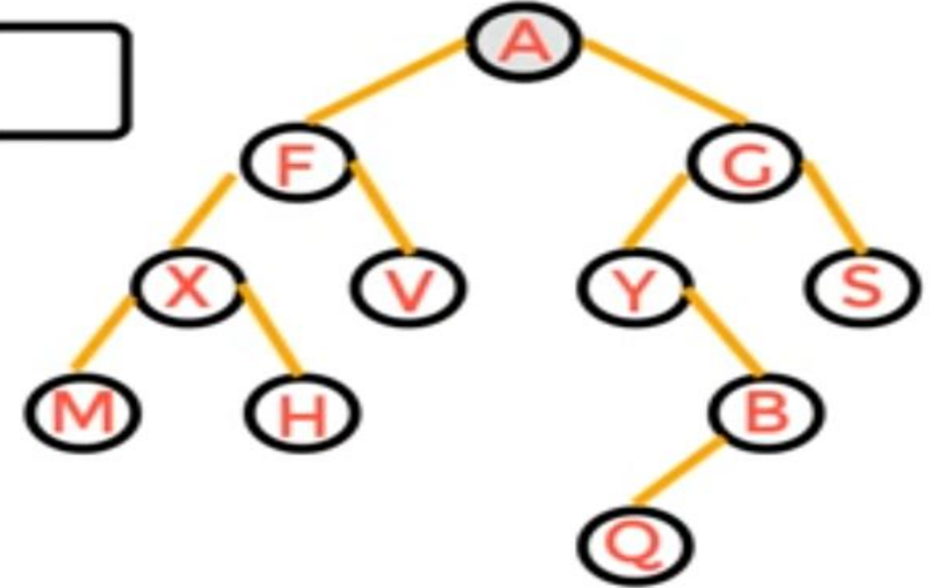


1) Create an empty queue q
2) Enqueue Root to q
3) Loop while q is not empty
   a) Print front of queue and remove it from queue.
   b) Enqueue node’s children (first left then right children) to q

التنقل بأسلوب level-order



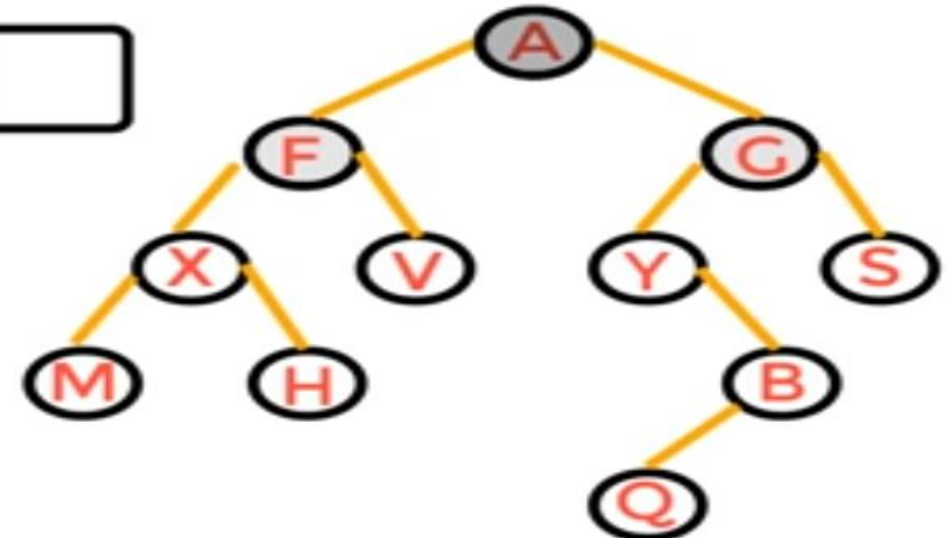
Queue





Queue

A



التنقل بأسلوب level–order



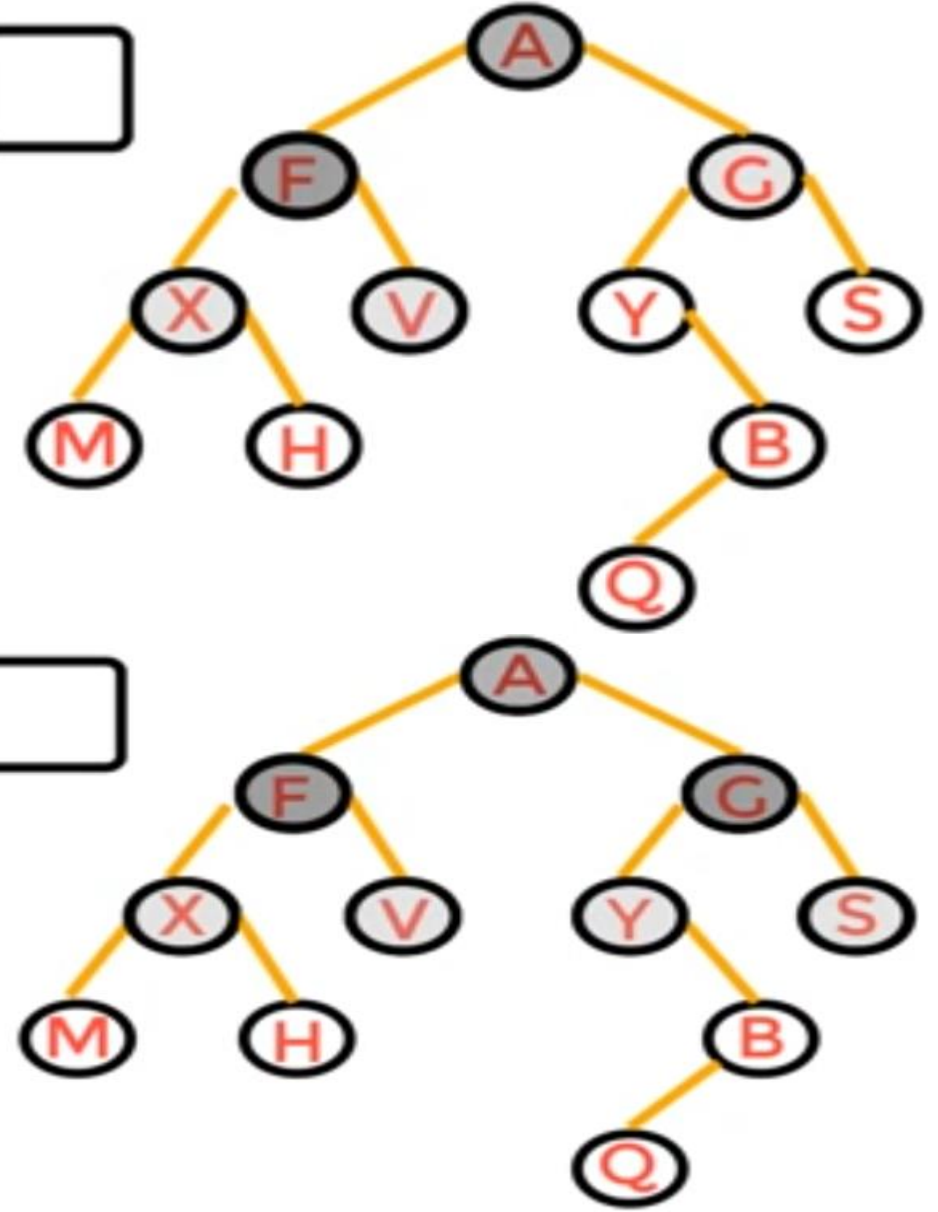
Queue

A F



Queue

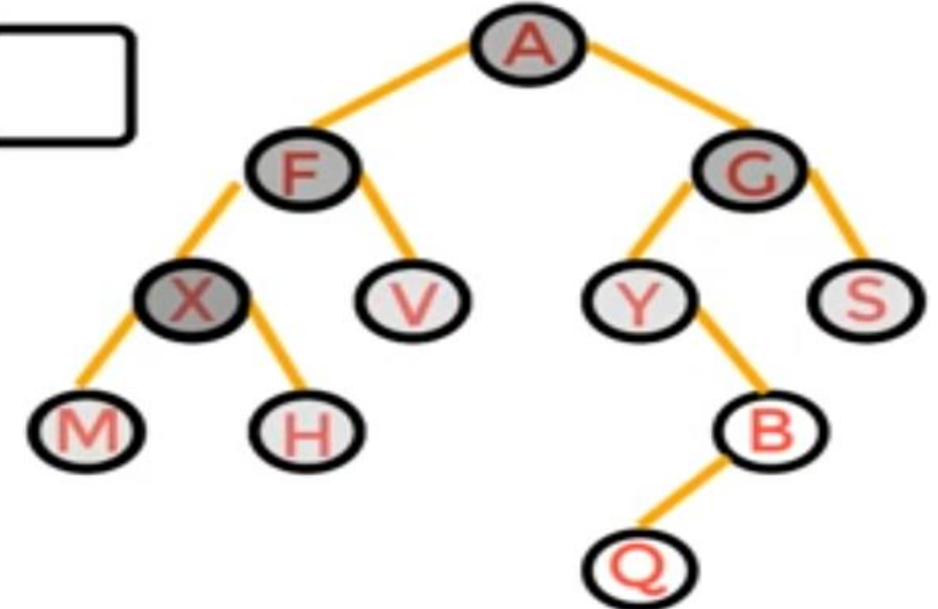
A F G



التنقل بأسلوب level-order



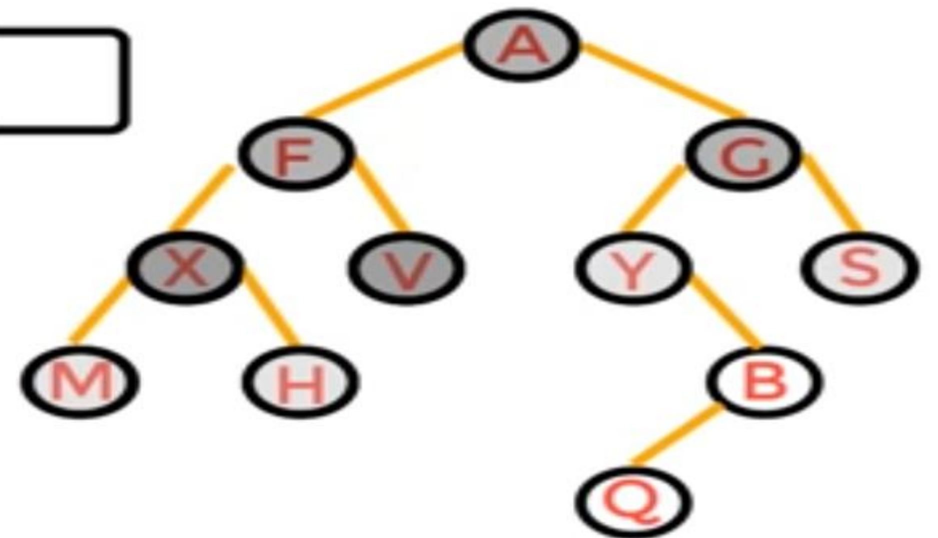
Queue



A F G X



Queue

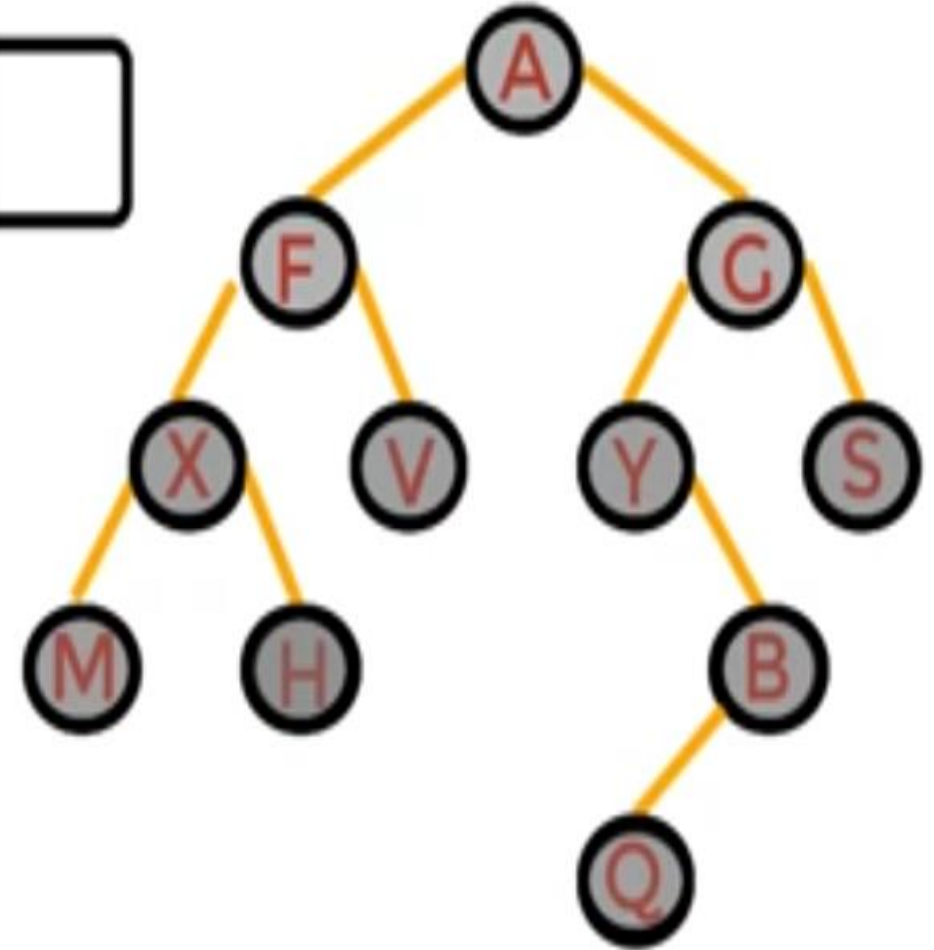


A F G X V

التنقل بأسلوب level–order



Queue

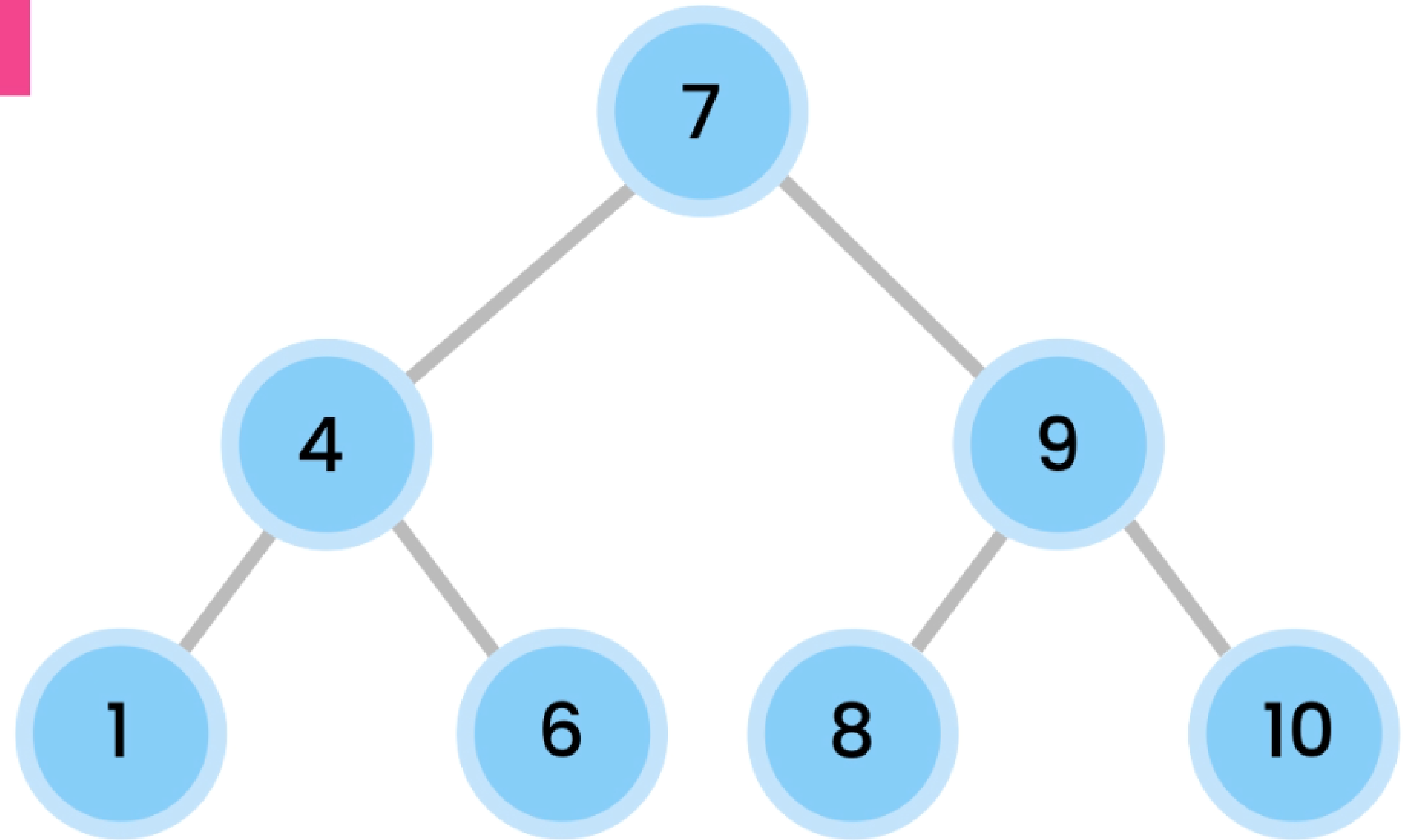


A F G X V Y S M H B Q

level–order التنقل بأسلوب

BREADTH FIRST

Level Order

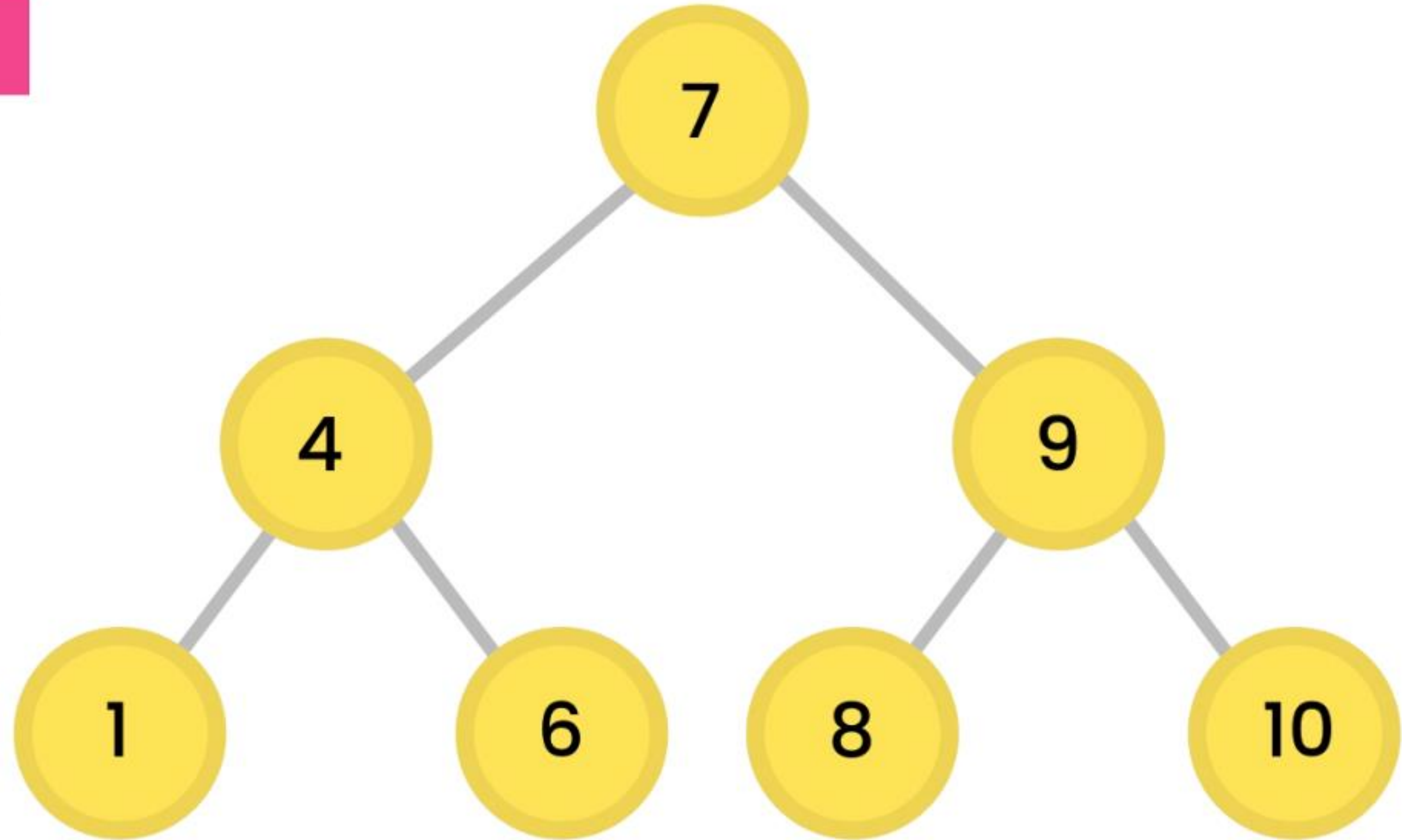


التنقل بأسلوب level–order

BREADTH FIRST

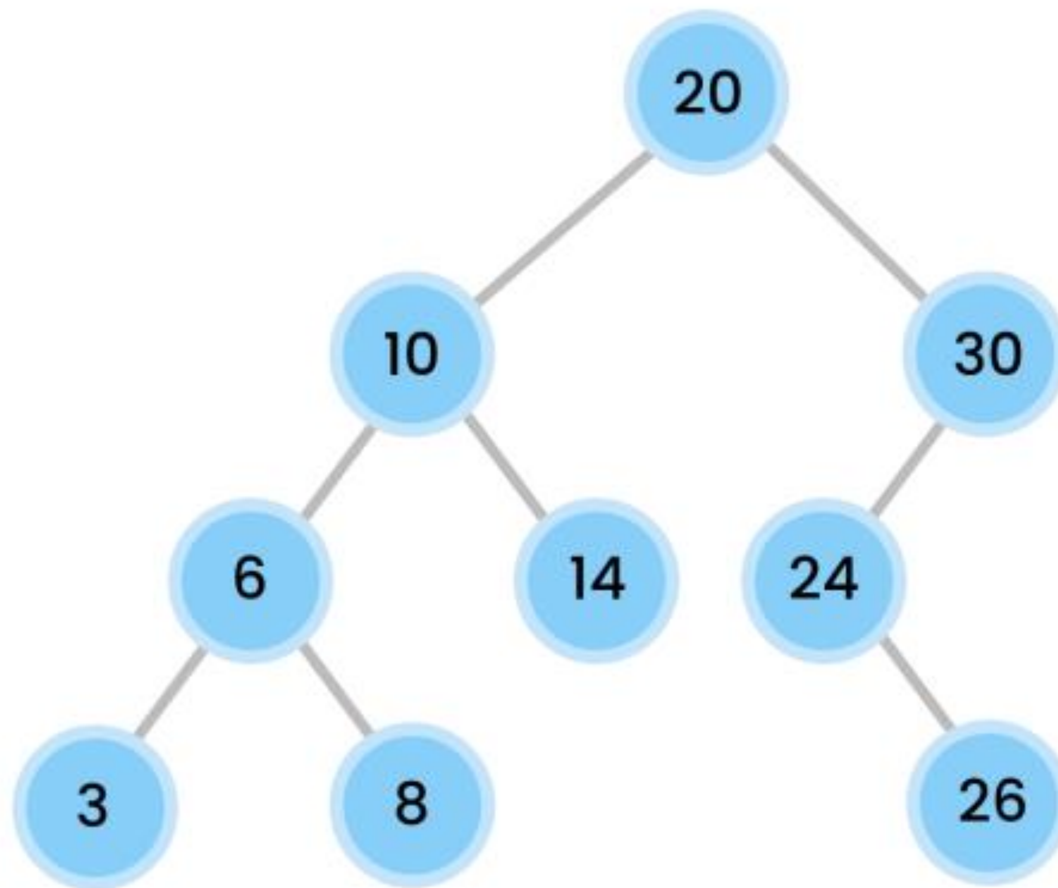
Level Order

7, 4, 9, 1, 6, 8, 10



تمرين

- مثال: ما نتيجة الطباعة عند تنفيذ أساليب التنقل على الشجرة الثنائية التالية



تمرين

- مثال: ما نتيجة الطباعة عند تنفيذ أساليب التنقل على الشجرة الثنائية التالية

- **Breadth First (Level Order):** 20, 10, 30, 6, 14, 24, 3, 8, 26
- **Depth First**
  - **Pre-order:** 20, 10, 6, 3, 8, 14, 30, 24, 26
  - **In-order:** 3, 6, 8, 10, 14, 20, 24, 26, 30
  - **Post-order:** 3, 8, 6, 14, 10, 26, 24, 30, 20

